CLUSTERING AND FEATURE DETECTION METHODS FOR HIGH-DIMENSIONAL DATA

A Dissertation
Presented to
The Academic Faculty

By

Geet Lahoti

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Industrial & Systems Engineering

Georgia Institute of Technology

August 2019

CLUSTERING AND FEATURE DETECTION METHODS FOR HIGH-DIMENSIONAL DATA

Approved by:

Dr. Chuck Zhang, Advisor
School of Industrial & Systems Engineering
Georgia Institute of Technology

Dr. Kamran Paynabar
School of Industrial & Systems Engineering
Georgia Institute of Technology

Dr. Jianjun (Jan) Shi
School of Industrial & Systems Engineering
Georgia Institute of Technology

Dr. Ben Wang
School of Industrial & Systems Engineering
Georgia Institute of Technology

Dr. Zhen Qian
Medical AI Lab
Tencent America, Palo Alto, CA

Date Approved: May 8, 2019

”Re-set, Re-adjust, Re-start, Re-focus...

As many times as you need to.”

Dedicated to my beloved parents, Kailash and Laxmi,

who inspire and motivate me constantly.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Professor Chuck Zhang, for his supervision, support, and encouragement throughout my Ph.D. journey. He has always motivated me to conduct impactful research. I would like to express my sincere appreciation to the committee members – Professor Kamran Paynabar, Professor Jianjun (Jan) Shi, Professor Ben Wang, and Dr. Zhen Qian, for their invaluable comments and suggestions on my Ph.D. research topics. I am also thankful to my former undergraduate advisors – Professor Manoj Kumar Tiwari and Professor Dilip Kumar Pratihar, for showing confidence in me and inculcating me with a desire to carry out research.

I would like to give special thanks to Dr. Chitta Ranjan, Jan Vlachy, and Jialei Chen for mentoring me throughout my Ph.D. studies. They have taught me how to define well-motivated research problems and develop novel methodologies. They have also taught me how to express scientific contributions and write research papers. They have also helped me with revisions of my research papers and presentations by carefully reading and commenting on them. They have always been available for research and non-research related discussions at any time. I would like to thank Dr. Hao Yan and Dr. Xiaowei Yue for providing me with advice and comments on my research work. I would like to thank Dr. Yung-Hang (Scott) Chang for helping me improve my programming skills. I would also like to thank my other research group members – Dr. Atiqur Rahman, Dr. Billyde Brown, Chin-Yuan Tseng, Hongzhen Tian, Dr. Kan Wang, Dr. Sang-Ha Hwang, Shancong Mou, and Dr. Zih-Huei Wang, for their support.

I want to thank my dear friends – Dr. Akhilesh Kumar, Dr. Anshuman Sinha, Dr. Arkadeep Kumar, Arvind Krishna, Digvijay Maheshwari, Dr. Ethan Mark, Fabien Caspani, Manasa Gudimella, Dr. Mohammed Nabhan, Dr. Priyabrata Mohapatra, Sachin Gupta, Sagar Agrawal, Dr. Samaneh Ebrahimi, Dr. Sandeep Samal, Dr. Satya Malladi, Dr. Tanmay Ghonge, Dr. Xiaolei Fang, and Yasaman Mohammad Shahi, who have always

supported me throughout this memorable Ph.D. journey.

I would like to express my deepest gratitude to my father, Kailash Lahoti, and my mother, Laxmi Lahoti for their endless support. They have always motivated me to reach great heights in my life. Their constant encouragement has helped me immensely during difficult times of the Ph.D. journey. I would like to thank my wife, Nitisha Jhawar, for her patience, selfless support and always believing in me. I would also like to thank my sister, Shubhangi Kakani, her husband, Yogesh Kakani, and her son, Medhansh Kakani; my father-in-law, Surendra Jhawar, my mother-in-law, Kiran Jhawar, and my sister-in-law, Muskan Jhawar, for always appreciating my efforts. Finally, I would like to thank the Almighty to make this dream a reality.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

In real-world scenarios, analyzing unlabeled, high-dimensional data is very important. This dissertation focuses on modeling and clustering censored spatio-temporal data, extracting medically-relevant features from computed tomography images of human heart, detecting anomalies from textured surface images, and detecting emotions from candidates' interview videos. In Chapter 1, these research problems and methodologies for solving these problems are briefly introduced.

The research topic presented in Chapter 2 focuses on clustering censored trajectory data. Trajectory data is widely found in advertisement and marketing domains. A trajectory is a spatio-temporal data instance in which an entity (e.g., service user) moves between a set of discrete states while spending a certain amount of time in each state. For example, on an online music streaming platform, a registered listener can be in the free tier, the subscription tier, and the state of inactivity. A trajectory of the listener will be her data stream denoting the state and the time spent in it. Analyzing the trajectory data helps understand user behavior, e.g., an online user's implicit intent to subscribe or unsubscribe to a service, which in turn can help monetize the service. However, a population of users can exhibit a variety of behaviors. Traditionally, users are targeted based on demographics to devise advertisement and marketing strategies. But, this leads to diluted targeting as two users of a particular region might have totally different tolerance toward a particular strategy. Therefore, clustering such a (heterogeneous) trajectory data can help segregate entities into different groups based on the behavior as reflected through their trajectories. Another challenge faced during the analysis of trajectory data is that most of the users are still active in the system when the observation period ends. This leads to incomplete (also known as censored) trajectories. In Chapter 2, we develop a novel unsupervised learning framework to find an unknown number of clusters within a given censored trajectory data set. The proposed framework is based on a mixture model-based clustering method that

utilizes a mixture of semi-Markov models to model the transition dynamics embedded in the trajectory data. While modeling each cluster using the semi-Markov model, the effect of censoring is also considered. Each user is assigned to a cluster based on its similarity to the cluster's profile. Cluster assignments and model parameters are simultaneously inferred using a robust expectation maximization-based algorithm which has the potential to overcome initialization issue and problem of finding an optimal number of clusters as faced in the case of traditional expectation maximization algorithm. The performance of the proposed framework is demonstrated using simulation and real case studies. In the simulation study, the proposed framework is found to outperform existing methods and in the real case study, the obtained clustering results are found to be effective for devising better advertisement and marketing strategies.

The research topic presented in Chapter 3 focuses on detecting pixel-level features in medical imaging data. Analyzing medical images such as computed tomographic images is very important for clinical decision making and surgery planning. For example, while planning medical treatment, it is critical to extract calcification and soft tissues from the pre-operative computed tomography image of a human heart. In the computed tomography image, calcification has higher pixel intensities than overall mean and soft tissues have lower pixel intensities. We term calcification as positive extreme features and soft tissues as negative extreme features. Identifying calcification is important as it is a pathological feature that has a connection to post-surgical complications, and extracting soft tissues is also critical as they represent organ morphology whose knowledge can help in pre-surgical planning. However, this is not an easy task because (a) conventional segmentation techniques require manual intervention and (b) existing automatic methods do not distinguish positive features from negative. In Chapter 3, we develop a novel, automatic image decomposition-based sparse extreme pixel-level feature detection model. The proposed model decomposes an image into mean and extreme features. A high-dimensional least squares regression with regularization and constraints is utilized to estimate model parameters. An efficient algo-

rithm based on the alternating direction method of multipliers and the proximal gradient method is developed to solve the optimization problem. Using simulation and real-world case studies, the effectiveness of the proposed model is elaborated. In the simulation study, it outperforms the benchmark methods and in the real case study, identified features are found to be of medical importance.

The research topic presented in Chapter 4 focuses on detecting sparse features in candidate interview videos and anomalies in textured surface images. In a candidate's video-based resume/interview, detecting sparse emotions is critical for the purpose of evaluation. Also, in manufacturing systems, detecting sparse anomalies in a product's image is essential for quality control. However, detecting features such as emotions and anomalies are challenging because (a) existing methods require processing the entire video frame by frame and the entire image patch by patch, which are time-consuming, (b) often video and image data are not quickly and completely available for server-side processing due to bandwidth reasons and increased data collection time, and (c) sometimes background and features are not easily distinguishable as their pixel intensities do not differ significantly, and there exists only a structural difference between them. In Chapter 4, we develop a novel sampling approach that employs an adaptive sampling method and a convolutional neural network model, to detect the sparse features of interest in high-dimensional input data. The adaptive sampling criterion explores the input data and exploits the regions of interest. The convolutional neural network model calculates the probability of the presence of desired features, which guides the exploitation component of the sampling strategy. The performance of the proposed approach is demonstrated using artificial and real data sets. The proposed approach reduces evaluation time and minimizes amount of input data to be accessed and processed while effectively identifying the desired features.

Finally, in Chapter 5, we conclude with major takeaways and scope for future work.

CHAPTER 1
INTRODUCTION AND BACKGROUND

The majority of the real-world data are unlabeled. Moreover, complex characteristics such as high-dimensionality and high variety pose significant analytical challenges. In statistical and machine learning, supervised and unsupervised methods are used to analyze labeled and unlabeled data, respectively. Compared to supervised learning methods, unsupervised learning is less developed. Therefore, this dissertation focuses on developing unsupervised methods to perform clustering and feature detection tasks in real-world high-dimensional data settings. Specifically, we develop methods to cluster censored spatio-temporal data, detect pixel-level features in medical imaging data, and adaptively detect anomalies in industrial optical inspection images and candidates emotions in interview videos.

The overarching objective of the unsupervised methods developed in this dissertation is to help stakeholders improve the performance of the associated systems through relevant analysis of the high-dimensional data. For example, (a) in an online service system such as internet radio, data-driven user behavior analysis can help service provider design better business strategies to maximize user engagement, (b) in a medical service system, automatic pathological feature detection from medical images can help surgeons plan medical surgery so that post-surgical complications can be minimized, (c) in a manufacturing system, automatic anomaly detection from product images can help a manufacturer control product quality, and (d) in an online interview practice platform, emotions detection from a candidate's interview video can help the platform owner in providing the user with evaluation and feedback.

In Section 1.1, different data types studied in this dissertation and challenges faced during their analysis are discussed. Section 1.2 details about the dissertation research topics and the developed methods. In Section 1.3, the structure of the dissertation is presented.

1.1 Unlabeled Data Types and Challenges

1.1.1 Trajectory Data

A trajectory is a spatio-temporal data instance in which an entity moves between a set of discrete states while spending a certain amount of time in each state. For example, on an online music streaming platform, a registered listener can be in the free tier, the subscription tier, and the state of inactivity. The trajectory of the listener will be her data stream denoting the state and the time spent in it. The trajectory data is widely found in fields such as marketing, healthcare, reliability, and web. It is considered important as analyzing such data can help understand entities' behavior, e.g., an online user's implicit intent to subscribe or unsubscribe to service; a patient's movement across different wards in a hospital; a mechanical component's degradation during its lifetime. However, there can be a variety of behaviors embedded in such data. Clustering such a (heterogeneous) trajectory data can help segregate entities into different groups based on the behavior as reflected through their transition dynamics. Another challenge is that most of the entities are still active in the system when the observation period ends thereby leading to censored or incomplete trajectories.

1.1.2 Medical Image Data

Analyzing medical images to detect pixel-level features is an essential but challenging task. Often times, a medical image contains multiple feature types – the ones with higher pixel intensities termed as positive (extreme) features and the others with lower pixel intensities as negative (extreme) features. For example, while planning medical treatment, it is important to identify, (a) calcification (a pathological feature which has connection to post-surgical complications) as positive features, and (b) soft tissues (organ morphology whose knowledge can help in pre-surgical planning) as negative features, from a pre-operative computed tomography image of human heart. However, this is not an easy task because

(a) conventional segmentation techniques require manual intervention and post-processing, and (b) existing automatic approaches do not distinguish positive features from negative.

1.1.3 Interview Video and Texture Surface Image Data

In online interview practice platforms, detecting sparse features in the form of emotions in a candidate's video is critical for evaluation. Also, in manufacturing systems, detecting sparse anomalies in a product's image is essential for quality control. However, these tasks are currently challenging because (a) existing methods require processing the entire video frame by frame and the entire image patch by patch, the time cost of which is often unacceptable, (b) often video and image data are not quickly and completely available for server-side processing due to bandwidth reasons and increased data collection time, and (c) sometimes background and features are not easily distinguishable as their pixel intensities do not differ significantly, and there exists only structural difference between them.

1.2 Dissertation Research Topics

To overcome the challenges faced during the analysis of the unlabeled data types discussed in Section 1.1, we propose different methods as detailed below:

1.2.1 Mixture of Semi-Markov Models-based Clustering Approach for Censored Trajectory Data

In this research topic, we propose a novel mixture model-based clustering methodology for finding an unknown number of possibly homogeneous clusters within a given censored trajectory data set. Each cluster is profiled using a semi-Markov model while considering the effect of censoring. Each sample entity is assigned to a cluster based on its similarity to the cluster's profile. Sample assignments and cluster profiles are simultaneously inferred using a robust expectation maximization algorithm. In the simulation study, our methodology demonstrates better performance than existing methods. The effectiveness of our

methodology is further elaborated using a real data set obtained from an internet music provider, where the obtained clustering results help devise better advertising and marketing strategies to target users in each cluster.

1.2.2 Image Decomposition-based Sparse Extreme Pixel-level Feature Detection Model

In this research topic, we propose a novel, automatic image decomposition-based sparse extreme pixel-level feature detection model to decompose an image into mean and extreme features. To estimate model parameters, a high-dimensional least squares regression with regularization and constraints is utilized. An efficient algorithm based on the alternating direction method of multipliers and the proximal gradient method is developed to solve the large-scale optimization problem. The effectiveness of the proposed model is demonstrated using synthetic tests and a real-world case study, where it shows a better performance than existing methods.

1.2.3 Convolutional Neural Network-assisted Adaptive Sampling for Sparse Feature Detection in Video and Image Data

In this research topic, we develop a novel sampling approach to detect the sparse features of interest in high-dimensional input data, e.g., the desired emotion in a video and the anomalies in an image. The approach employs an adaptive sampling technique coupled with a convolutional neural network model. The adaptive sampling criterion smartly explores the high-dimensional input and exploits the regions of interest. The convolutional neural network model determines the likelihood of the presence of desired features, which guides the exploitation component of the sampling strategy. The effectiveness of the proposed approach is illustrated using artificial and real data sets. The proposed approach reduces the feature detection time and minimizes the amount of input data to be accessed and processed while effectively identifying the desired features.

The overall dissertation overview is presented in Figure 1.1. It summarizes the data-

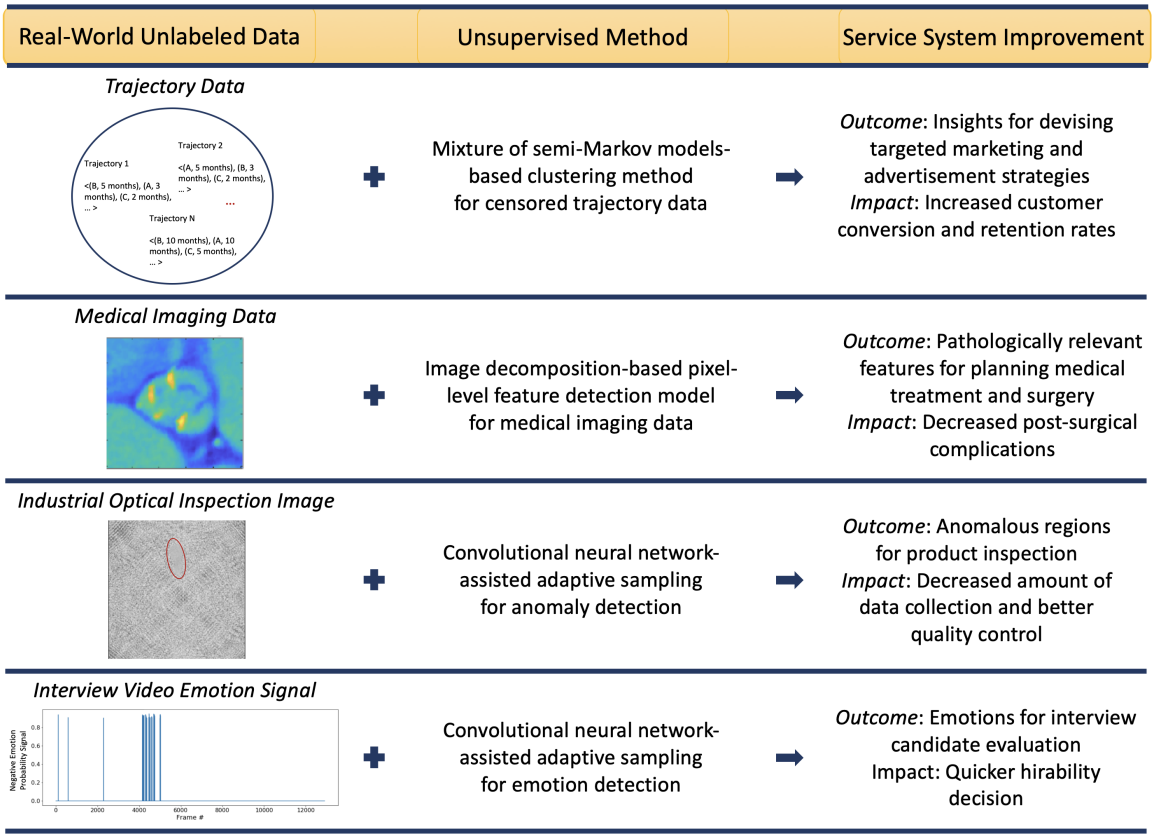

Figure 1.1: Dissertation overview – unlabeled data types, developed unsupervised learning methods, and system performance improvement (overarching goal)

types handled in this dissertation, the methods developed to tackle those data, the outcome produced by analyzing the data, and the potential impact in the form of service system improvement.

1.3 Dissertation Organization

This dissertation is organized as follows: Chapter 2 describes the developed unsupervised learning framework to cluster censored trajectory data. Chapter 3 introduces the developed pixel-level feature detection model to extract pathological features from medical images. Chapter 4 explains the developed sampling approach to detect anomalies from textured surface images and emotions from candidate interview videos. Finally, in Chapter 5, conclusions and scope for future research are presented.

CHAPTER 2

MIXTURE OF SEMI-MARKOV MODELS-BASED CLUSTERING APPROACH FOR CENSORED TRAJECTORY DATA

2.1 Introduction

A trajectory is a spatio-temporal data instance in which an entity moves between a set of discrete states while spending a certain amount of time in each state. For example, on an online music streaming platform, a registered listener can be in the *free tier* (F), the *subscription tier* (S), or the *state of inactivity* (I). A trajectory of listener will be her data stream denoting the state and the time spent in it, e.g., <(F, 5 months), (S, 3 months), (I, 2 months),...> means the listener is in the free state for 5 months, then moves to the subscription state and stays there for 3 months, followed by going to the inactivity state for 2 months, and so on. We will call a sample of such trajectories as tier-transition (TT) data, in the rest of the paper. Trajectory data is also found in other settings. For instance, most of session logs data are a trajectory, such as in a music listening session on an online radio, the states can be *play list*, *radio*, *browsing*, among others, and a trajectory will be the sequence of these states and the time spent there. Similar trajectories can be drawn from user data on Netflix, where the states can be the genre of movies/series, and a user trajectory defined as the amount of time she spent watching one genre before moving to another. Another example can be seen in the domain of healthcare, e.g., disease like cancer has several stages and a patient may undergo different stages before she fully recovers or dies. Here, the states correspond to the stages of the disease and a trajectory can be defined as the sequence of stages and time spent in each of those stages. Figure 2.1 shows a generalized state transition diagram, where there are s_1 transient states and an absorbing state.

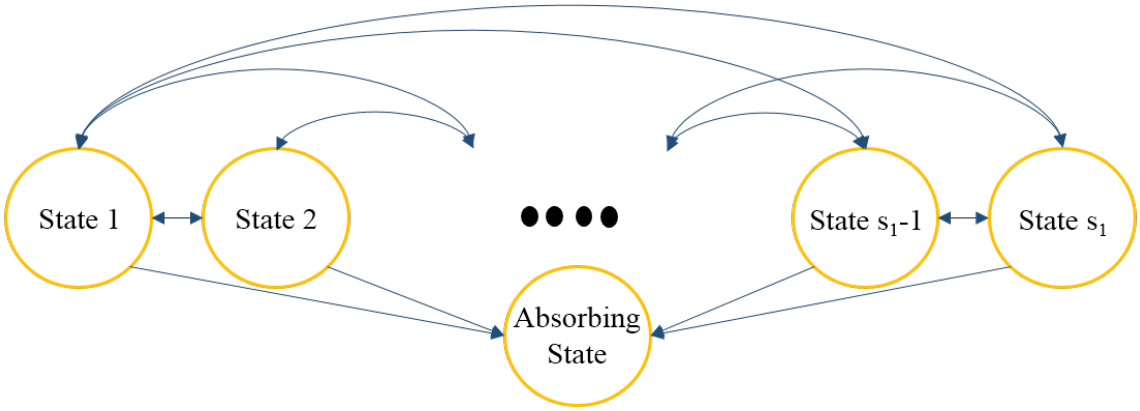

Figure 2.1: Generalized state transition diagram

Understanding the behavior of the entities (listeners on internet radio, users on Netflix, patients of a multi-stage disease) is extremely important for tasks such as advertisements, sales promotions, and better services. A behavior is defined based on the problem at hand. For example, in the TT data, the behavior characterizes a user's implicit intent to subscribe or unsubscribe to the service after spending some time in any of the remaining states. If we understand a user's behavior to foretell her reactions (moving to another state) to our actions, then we can improve tasks such as advertisements and sales promotions through superior targeting. Considering trajectory as the first-hand proxy for a user's behavior, one can uncover underlying behavior by analyzing the trajectories of the users.

Multi-state models can help model the behavior embedded in a trajectory data set. Markov and semi-Markov models are popular model choices. These models follow the well-known Markov property - the future state depends only on the current state. Semi-Markov models allow greater flexibility by making the transition from one state to another depend not only on the current state but also the time spent in the current state. Semi-Markov models are used to model the behavior in various domains such as reliability, healthcare, marketing, and travel. In such models, transition dynamics are governed by transition probability and holding time distributions.

However, there can be a variety of behaviors in a population of entities. For instance, in

the TT data, the tendency of a music lover to move from free tier to subscription would be higher than that for a regular user who is not so much music enthusiast but prefers trying different services. This leads to the presence of heterogeneous behaviors in the population.

Traditionally, heterogeneity among users is addressed by demographic targeting. For example, up-selling subscription tier to a particular user demographics. This is more in line of stereotyping users and assuming users with the same demographics will have the same behavior/response. However, two males of the same age and the same city can have entirely different tolerance to advertisements, and the optimal advertising mix for them can be different. As observed in [1], methods based on this demographic premise are not as effective. This is because the generalization leads to diluted targeting. We can target more effectively if we adjust depending on the actual behavior type.

Another way to tackle heterogeneity could be to make transition probability and holding time distributions of the semi-Markov model a function of various attributes such as sex, education, and geographical region. This could help model the heterogeneity embedded in the trajectory data. However, it is worth noting that not all attributes can be considered and measured. Most of the factors leading to heterogeneity are probably unobservable. Therefore, it is beneficial to cluster such a heterogeneous population of entities' trajectories using a mixture of models rather than a single model.

One other challenge is the presence of censoring. It is not always possible to observe the complete trajectories. When a group of users is selected to study their behavior, it might happen that some users are still in the system and their trajectories are incomplete, as illustrated in Figure 2.2. In such scenarios, it becomes imperative to consider the effect of censoring in the model.

To address the aforementioned challenges, this work develops a mixture of models-based approach for clustering the heterogeneous entities' behavior embedded in the trajectory data while explicitly accounting for censoring. The proposed approach utilizes a mixture of discrete-time semi-Markov models, where distributions of mixture components

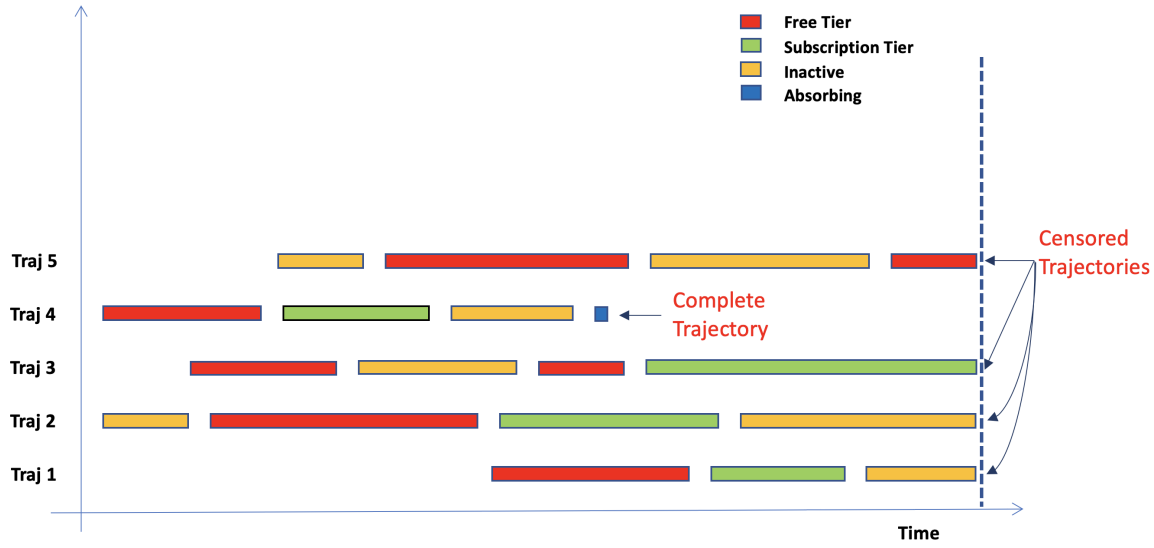

Figure 2.2: Sample uncensored and censored trajectories

are estimated simultaneously for several subgroups of a heterogeneous population while also estimating each entity's cluster assignment. The clusters are not known a priori but are obtained by leveraging the similarities in their behavior. The number of clusters depends on the extent of heterogeneity present in the population. An expectation-minimization (EM) – based algorithm is proposed to estimate the parameters of the mixture model, cluster assignments, and adequate number of clusters. The proposed approach is tested on simulated data sets as well as real data obtained from an internet radio service provider.

The outline of the paper is as follows. Problem background and definition have been put forth in this section. Literature is reviewed in Section 2. In Section 3, the mixture model is developed. The model estimation procedure is detailed in Section 4. A simulation study is presented in Section 5. The applicability of the proposed method is corroborated using a real case study in Section 6.

2.2 Related Work

In this section, prior work related to clustering is reviewed. The objective of clustering is to explore data to uncover the embedded latent structure and summarize it in a limited

number of clusters. Data in each cluster turns out to be more homogeneous. Mainly, there are four categories of clustering algorithms: hierarchical clustering, distance-based clustering, topographic clustering, and model-based clustering.

Agglomerative and splitting hierarchical clustering methods [2] build a hierarchy of clusters. Most popular distance-based clustering algorithm is K-means algorithm [3, 4]. K-means algorithm, an iterative procedure, aims at partitioning a given data set into a number of clusters defined a priori by minimizing the within-cluster variance and maximizing the between-cluster variance. Fuzzy K-means [5] and trimmed K-means [6, 7] are some of the variants of K-means. Self-organizing map [8], an unsupervised neural-based approach, is one of the most popular topographic clustering methods. These clustering methods can be regarded as deterministic as they do not use a density model for the data.

Finite mixture model is a popular and successful model-based clustering approach based on density modeling [9, 10, 11, 12]. It provides a flexible probabilistic framework for clustering. In this approach, the data probability density function is assumed to consist of a mixture of components where each component density is associated with a cluster. The objective, therefore, becomes estimating the parameters of the mixture model. This can be accomplished using the well-known expectation-maximization (EM) algorithm [13]. Due to its desirable properties of stability and reliable convergence, the EM algorithm is heavily employed in the mixture modeling framework. For more details on EM, please refer to [14]. In the model-based clustering approaches, the optimal number of clusters can be decided after the learning process using some information criteria for model selection, such as BIC [15].

Generally, the model-based clustering techniques deal with data living in Euclidean spaces. Gaussian mixture models in conjunction with EM algorithm are prominently used to model such data [9, 14, 16]. However, in many scenarios, the data are curves, functions, time series or trajectories, instead of vectors. In such cases, the data can be modeled using suitable statistical models and the clustering can be performed using a mixture of such

models. Some of the examples in this category are: polynomial regression mixtures, splines and B-splines regression mixtures, and generative polynomial piecewise regression [17, 18, 19, 20]. The parameters are still estimated by maximizing the observed-data log-likelihood through the EM algorithm.

Researchers have also developed methods to deal with the problem of clustering the trajectory data. In [21], researchers analyzed the discrete demographic sequential data using a finite mixture of Markov chains to identify subgroups of women with similar contraceptive use patterns. However, Markov models are limited to modeling the spatial variation present in the trajectory data. They don't have the explicit capability to tackle the temporal variation. In [1], the authors presented a clustering and scheduling integrated approach for patient flow modeling. They developed a novel semi-Markov model-based clustering scheme to cluster patients according to the similarity in their trajectories and estimate cluster trajectory distributions which act as an input for scheduling optimization approaches. Semi-Markov models can not only model the spatial variation in the trajectory data but also the temporal variation.

Although semi-Markov models are able to capture both spatial and temporal variations, they can't be used in their original form if the trajectory data for some samples of the population is not realized completely. This may occur due to either a subject leaving the study or the study ending before the event has taken place. This type of mechanism is known as the right censoring. In [22], researchers presented a semi-Markov model for the analysis of partially right-censored multi-state data. Model unknowns were estimated using non-parametric likelihood methods. In [23], researchers reformulated the semi-Markov model presented by [22] and derived the non-parametric maximum likelihood estimators for the model unknowns, in terms of hazard and Kaplan & Meier [24] survival estimators.

In this work, we develop an unsupervised learning approach for clustering censored trajectory data. Particularly, a model-based clustering framework is developed to group the trajectory data. The proposed approach deals with the data where the observations are

trajectories rather than vectors as in a multivariate Gaussian mixture model [16] or curves as in a regression mixture model [25]. It utilizes a mixture of semi-Markov models to deal with the trajectory clustering problem. Each mixture component, semi-Markov model, captures the spatio-temporal pattern embedded in the trajectories and explicitly accounts for censoring in the trajectory data. An EM-based algorithm is proposed to estimate the parameters of the mixture of models. The proposed algorithm has the potential to tackle the problem of initialization. It also has the capability to select the optimal number of clusters. In the next section, the proposed model and algorithm are discussed.

2.3 Clustering Model

In this section, a clustering model to group partially or fully censored trajectories is presented. Particularly, a model-based clustering approach is adopted. We assume that the trajectories are generated from a mixture of semi-Markov models. The effect of censoring is considered while modeling the trajectories using the semi-Markov models. While estimating each of the mixture components, i.e., the distributions of the semi-Markov models in the mixture, a cluster assignment for each sample in the population is also estimated.

2.3.1 Mixture of Semi-Markov Models with Censoring (MoSMMC)

Let K' be the set of unknown clusters within a given population, where each cluster follows a unique semi-Markov process. The population of trajectory data, thus, follows a mixture of an unknown number of semi-Markov processes equal to $|K'|$. Each mixture component, which we term as cluster henceforth, has a different semi-Markov process distribution. An entity trajectory is denoted by l ($l = 1, ..., L$). Let $E = \{\underline{E}, \overline{E}\}$ be the set of all possible states representing the movement of an entity, where $\underline{E}$ is the set of all transient states and $\overline{E}$ is the set of all absorbing states.

Entities within each cluster follow homogeneous first-order semi-Markov model with s total states, the first s_1 of which are transient. For each entity, Z_0 denotes the initial

state, Z_n denotes the state corresponding to the n^{th} event, and T_n represents the sojourn time between the $(n-1)^{st}$ and n^{th} events. History for an entity l can be denoted by $H_l = (Z_0, T_1, Z_1, T_2, Z_2, ..., T_m, Z_m)$. Length m varies over subjects. Complete history ends with a transition to an absorbing state $(s_1 < Z_m \leq s)$, whereas a right-censored history provides only a lower bound for T_m and is denoted by $Z_m = s + 1$.

We proceed with the problem formulation by defining a set of parameters, $\boldsymbol{\Theta} = \{\Theta^{(k')}\}$, $k' \in K'$, where $\Theta^{(k')}$ comprises of mixture weight, $\pi^{(k')}$, and semi-Markov process parameters, $\{\boldsymbol{\rho}^{(k')}, \boldsymbol{\theta}^{(k')}, \mathbf{Q}^{(k')}\}$, for the k'-th mixture. The mixture weight, $\pi^{(k')}$, indicates the probability of an entity belonging to cluster k'. Letting X_l be a hidden variable representing the cluster index for an entity l, then the mixture weight can be written as, $\pi^{(k')} = pr_{\Theta}(X_l = k')$, and the constraint, $\sum_{k' \in K'} \pi^{(k')} = 1$, holds. The initial state probability is denoted by $\boldsymbol{\rho}^{(k')} = \{\rho_i^{(k')}\}$, $i \in \underline{E}$, where $\rho_i^{(k')} = p_{\Theta}(Z_0 = i \,|\, X_l = k')$ represents the probability of the first state of an entity's trajectory from cluster k' being i. The matrix, $\boldsymbol{\theta}^{(k')} = [\theta^{(k')}(i,\, j)]$, $i \in \underline{E}$, $j \in E$, denotes the transition probability matrix, where $\theta^{(k')}(i,\, j) = p_{\Theta}(Z_{n+1} = j \,|\, Z_n = i,\, X_l = k')$ indicates the probability of transitioning from state i to j for an entity l in cluster k'. Since $\{\boldsymbol{\rho}^{(k')},\, \theta^{(k')}(i,\, \bullet)\}$ are probability distributions, $\sum_{i \in \underline{E}} \rho_i^{(k')} = 1$ and $\sum_{j \in E} \theta^{(k')}(i,\, j) = 1$ hold. Finally, $\mathbf{Q}^{(k')} = [Q^{(k')}(t;\, i, j)]$, $i \in \underline{E}$, $j \in E$, is a three-dimensional tensor representing the sojourn time probability distribution, where $Q^{(k')}(t;\, i, j) = p_{\Theta}(T_{n+1} > t | Z_n = i, Z_{n+1} = j, X_l = k')$ is the probability that the time spent in state i before transitioning to j for an entity in cluster k' is greater than t.

Suppose $\upsilon_1 < ... < \upsilon_k < ... < \upsilon_M$ indicate the M distinct uncensored sojourn times observed and suppose $m_{ijk}^{k'}$ denote the number of uncensored sojourn times from a state i to a state j of a length υ_k for $i = 1, ..., s_1$; $j = 1, ..., s$; $k = 1, ..., M$; and $k' = 1, ..., K'$. Also, suppose $m_{i,s+1,k}^{k'}$ indicate the number of sojourn times in a state i that are censored in $[\upsilon_k, \upsilon_{k+1})$, where $k = 0, 1, ..., M$, $\upsilon_0 = 0$, and $\upsilon_{M+1} = \infty$. For fixed i and k', the nonparametric maximum likelihood solution places mass only at the υ_k's, and if the largest time

in state i is censored, it places mass in the open interval (υ_M, ∞) as well [22, 24]. Therefore, each T_n can be considered a discrete random variable. The discrete event-specific hazard function is defined as follows:

$$\lambda_{ijk}^{(k^{'})} = p(T_{n+1} = \upsilon_k, Z_{n+1} = j | T_{n+1} \geq \upsilon_k, Z_n = i, X_l = k^{'}). \tag{2.1}$$

Also, the survival function for sojourn times in a state i for a cluster $k^{'}$ is defined as follows:

$$S^{(k^{'})}(t;\, i) = p(T_{n+1} > t | Z_n = i, X_l = k^{'}). \tag{2.2}$$

Also, the conditional probability that a sojourn time in a state i exceeds υ_k given that it exceeds υ_{k-1}, is given by:

$$q_{ik}^{(k^{'})} = \frac{S^{(k')}(\upsilon_k; i)}{S^{(k')}(\upsilon_{k-1}; i)} = 1 - (\lambda_{i1k}^{(k^{'})} + ... + \lambda_{isk}^{(k^{'})}) = 1 - \sum_{j=1}^{s} \lambda_{ijk}^{(k^{'})}. \tag{2.3}$$

Contribution to the likelihood for an uncensored transition to a state j after spending a length υ_k in a state i for a cluster $k^{'}$ is $\lambda_{ijk}^{(k^{'})} S^{(k^{'})}(\upsilon_{k-1};\, i)$ and the likelihood contribution for a sojourn time censored in $[\upsilon_k, \upsilon_{k+1})$ in a state i for a cluster $k^{'}$, is $S^{(k^{'})}(\upsilon_k;\, i)$. Next, we express the conditional probability of l^{th} entity's history, H_l, given it is generated by cluster $k^{'}$ in the following manner:

$$\begin{aligned}
pr_{\boldsymbol{\Theta}}(H_l \,|\, X_l = k') &= pr(Z_{l,0} = z_{l,0} | \theta^{(k')}) \left\{ \prod_{n=1}^{m^{(l)}-1} (C_{l,n}^{E,(k')}) \right\} (C_{l,m^{(l)}}^{E,(k')})^{\delta(s - z_{l,m^{(l)}})} \times \\
&\quad (C_{l,m^{(l)}}^{R,(k')})^{\delta(z_{l,m^{(l)}} - s - 1)} \\
&= \rho_{z_{l,0}}^{(k')} \left[\prod_{n=1}^{m^{(l)}-1} \left\{ \lambda_{z_{l,n-1}, z_{l,n}, t_{l,n}}^{(k')} S^{(k')}(t_{l,n}^{-1}; z_{l,n-1}) \right\} \right] \times \\
&\quad \left\{ \lambda_{z_{l,m^{(l)}-1}, z_{l,m^{(l)}}, t_{l,m^{(l)}}}^{(k')} S^{(k')}(t_{l,m^{(l)}}^{-1}; z_{l,m^{(l)}-1}) \right\}^{\delta(s - z_{l,m^{(l)}})} \times \\
&\quad \left\{ S^{(k')}(\upsilon_k :\, \upsilon_k \leq t_{l,m^{(l)}} < \upsilon_{k+1},\, k = 0, 1, ..., M;\, z_{l,m^{(l)}-1}) \right\}^{\delta(z_{l,m^{(l)}} - s - 1)},
\end{aligned}$$

where $\delta(a) = 1$ if $a \geqslant 0$ and zero otherwise. Next, we write the probability distribution function of the semi-Markov models mixture with $K^{'}$ components as shown below:

$$\begin{aligned} pr(H_l \,|\, \mathbf{\Theta}) &= \sum_{k' \in K'} pr_{\mathbf{\Theta}}(X_l = k')\, p_{\mathbf{\Theta}}(H_l \,|\, X_l = k') \\ &= \sum_{k' \in K'} \pi^{(k')} \Bigg[pr(Z_{l,0} = z_{l,0} | \theta^{(k')}) \left\{ \prod_{n=1}^{m^{(l)}-1} (C_{l,n}^{E,(k')}) \right\} \times \\ &\quad (C_{l,m^{(l)}}^{E,(k')})^{\delta(s - z_{l,m^{(l)}})} (C_{l,m^{(l)}}^{R,(k')})^{\delta(z_{l,m^{(l)}} - s - 1)} \Bigg]. \end{aligned}$$

The likelihood function for a given i.i.d. sample of L trajectories, $\mathbf{H} = \{H_l;\, l = 1, ..., L\}$, is given by:

$$\begin{aligned} pr_{\mathbf{\Theta}}(\mathbf{H}) &= \prod_{l=1}^{L} pr(H_l \,|\, \mathbf{\Theta}) \\ &= \prod_{l=1}^{L} \Bigg[\sum_{k' \in K'} \pi^{(k')} \Bigg[pr(Z_{l,0} = z_{l,0} | \theta^{(k')}) \left\{ \prod_{n=1}^{m^{(l)}-1} (C_{l,n}^{E,(k')}) \right\} \times \\ &\quad (C_{l,m^{(l)}}^{E,(k')})^{\delta(s - z_{l,m^{(l)}})} (C_{l,m^{(l)}}^{R,(k')})^{\delta(z_{l,m^{(l)}} - s - 1)} \Bigg] \Bigg], \end{aligned} \tag{2.4}$$

and the log-likelihood function is given as follows:

$$\begin{aligned} \mathcal{L}(\mathbf{\Theta}) &= \sum_{l=1}^{L} log \Bigg[\sum_{k' \in K'} \pi^{(k')} \Bigg[pr(Z_{l,0} = z_{l,0} | \theta^{(k')}) \left\{ \prod_{n=1}^{m^{(l)}-1} (C_{l,n}^{E,(k')}) \right\} \times \\ &\quad (C_{l,m^{(l)}}^{E,(k')})^{\delta(s - z_{l,m^{(l)}})} (C_{l,m^{(l)}}^{R,(k')})^{\delta(z_{l,m^{(l)}} - s - 1)} \Bigg] \Bigg]. \end{aligned} \tag{2.5}$$

2.4 Model Estimation Procedure

In order to estimate the parameters of the mixture model, we can maximize the log-likelihood function in (2.5). However, a closed-form solution for the parameter estimators is not feasible to achieve. Furthermore, the log-likelihood function is non-convex so standard convex

optimization methods cannot be used for the estimation. We could adopt the standard EM algorithm. However, the EM algorithm suffers from the initialization issue and also, the number of sub-groups must be provided a priori. Therefore, we propose an EM-based algorithm for model-based censored trajectory clustering using the mixture of semi-Markov models. The current work is in the same spirit as the EM-like algorithm from [16]. Here, their idea is extended to the case of trajectory data clustering. The proposed EM-based algorithm is able to deal with the initialization issue. Also, it can select the optimal number of subgroups. The procedure allows for fitting the semi-Markov mixture model without using a separate algorithm for initializing the EM algorithm and without specifying a specific number of subgroups in the population a prior. Next, we derive the proposed estimation approach.

2.4.1 Penalized Likelihood

Instead of using the standard observed-data log-likelihood in (2.5), we maximize the following penalized log-likelihood criterion:

$$\mathcal{J}(\alpha, \mathbf{\Theta}) = \mathcal{L}(\mathbf{\Theta}) - \alpha H(\mathbf{X}), \tag{2.6}$$

where $\alpha \geq 0$ is the regularization parameter which controls the complexity of the model and $H(\mathbf{X}) = -\sum_{l=1}^{L} \sum_{k'=1}^{K'} \pi^{(k')} \log \pi^{(k')}$ is the entropy for the hidden variables.

2.4.2 Parameter estimation via robust expectation-maximization (REM) algorithm

Given an i.i.d set of trajectories, we iteratively maximize the penalized log-likelihood function in (2.6) using the robust EM algorithm for model-based trajectory clustering. Before we discuss the EM steps, the penalized complete-data log-likelihood, which is the foundation for EM formulation, is presented as follows:

$$\mathcal{J}_c(\alpha, \mathbf{\Theta}) = \sum_{l=1}^{L} \sum_{k' \in K'} X_{lk'} \log \Bigg[\pi^{(k')} pr(Z_{l,0} = z_{l,0} | \theta^{(k')}) \left\{ \prod_{n=1}^{m^{(l)}-1} C_{l,n}^{E,(k')} \right\} \times$$
$$(C_{l,m^{(l)}}^{E,(k')})^{\delta(s - z_{l,m^{(l)}})} (C_{l,m^{(l)}}^{R,(k')})^{\delta(z_{l,m^{(l)}} - s - 1)} \Bigg]$$
$$+\alpha \sum_{l=1}^{L} \sum_{k'=1}^{K'} \pi^{(k')} \log \pi^{(k')}, \tag{2.7}$$

where $X_{lk'}$ is a binary-valued variable such that $X_{lk'} = 1$ if $X_l = k'$ (i.e., if the l^{th} trajectory is generated from the k'-th mixture component) and $X_{lk'} = 0$ otherwise. Starting with an initial solution, the proposed algorithm alternates between the following two steps until convergence.

E-step

In this step, given the observed trajectory data $\mathbf{H}$ and the current parameter estimate $\mathbf{\Theta}^{(p)}$, where p is the iteration counter, the expectation of the penalized complete-data log-likelihood in (2.7) is computed over the hidden variable $\mathbf{X}$ as shown below:

$$\begin{aligned}
\mathcal{Q}(\alpha, \mathbf{\Theta} | \mathbf{\Theta}^{(p)}) &= \mathbb{E}_{\mathbf{\Theta}^{(p)}} [\mathcal{J}_c(\alpha, \mathbf{\Theta}) | \mathbf{H}] \\
&= \sum_{l=1}^{L} \sum_{k' \in K'} \mathbb{E}_{\mathbf{\Theta}^{(p)}} [X_{lk'} | \mathbf{H}] \log \left\{ \pi^{(k')} pr_{\mathbf{\Theta}}(H_l \,|\, X_l = k') \right\} + \\
&\quad \alpha \sum_{l=1}^{L} \sum_{k'=1}^{K'} \pi^{(k')} \log \pi^{(k')} \\
&= \sum_{l=1}^{L} \sum_{k' \in K'} \Omega_{lk'}(\mathbf{\Theta}^{(p)}) \log \left\{ \pi^{(k')} pr_{\mathbf{\Theta}}(H_l \,|\, X_l = k') \right\} + \\
&\quad \alpha \sum_{l=1}^{L} \sum_{k'=1}^{K'} \pi^{(k')} \log \pi^{(k')},
\end{aligned} \tag{2.8}$$

where

$$\Omega_{lk'}(\boldsymbol{\Theta}^{(p)}) \; = pr(X_l = k' \,|\, H_l,\, \boldsymbol{\Theta}^{(p)}) \; = \frac{\pi^{(k')} pr_{\boldsymbol{\Theta}^{(\mathbf{p})}}(H_l \,|\, X_l = k')}{\sum_{k' \in K'} \pi^{(k')}\, pr_{\boldsymbol{\Theta}^{(\mathbf{p})}}(H_l \,|\, X_l = k')} \tag{2.9}$$

is the posterior probability that the trajectory l is generated from the $k^{'}$-th cluster. Hence, in this step, only the computation of the posterior probabilities $\Omega_{lk'}$ where, $l = 1, ..., L, k^{'} = 1, ..., K^{'}$, is required.

M-step

In this step, the parameters are updated by maximizing the $\mathcal{Q}$-function in (2.8) with respect to $\boldsymbol{\Theta}$ as shown below:

$$\boldsymbol{\Theta}^{(p+1)} \; = \; \arg\max_{\boldsymbol{\Theta}} \mathcal{Q}(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{(p)}). \tag{2.10}$$

The $\mathcal{Q}$-function in (2.8) can be decomposed as shown below:

$$\mathcal{Q}(\alpha, \boldsymbol{\Theta}|\boldsymbol{\Theta}^{(p)}) = \mathcal{Q}_{\pi}(\alpha, \pi^1, ..., \pi^{K^{'}}, |\boldsymbol{\Theta}^{(p)}) + \mathcal{Q}_{-\pi}(\alpha, \{\boldsymbol{\rho}^{(k^{'})},\, \boldsymbol{\theta}^{(k^{'})},\, \mathbf{Q}^{(k^{'})}\}_{k^{'}=1}^{K^{'}}|\boldsymbol{\Theta}^{(p)}), \tag{2.11}$$

where

$$\mathcal{Q}_{\pi}(\alpha, \pi^{(1)}, ..., \pi^{(K^{'})}, |\boldsymbol{\Theta}^{(p)}) = \sum_{l=1}^{L} \sum_{k' \in K'} \Omega_{lk'}(\boldsymbol{\Theta}^{(p)}) \; \log \; \pi^{(k')} + \alpha \sum_{l=1}^{L} \sum_{k'=1}^{K^{'}} \pi^{(k')} \; \log \; \pi^{(k')}, \tag{2.12}$$

and

$$\begin{aligned}
\mathcal{Q}_{-\pi}(\alpha, \{\boldsymbol{\rho}^{(k')}, \boldsymbol{\theta}^{(k')}, \mathbf{Q}^{(k')}\}_{k'=1}^{K'} | \boldsymbol{\Theta}^{(p)}) &= \sum_{l=1}^{L} \sum_{k' \in K'} \Omega_{lk'}(\boldsymbol{\Theta}^{(p)}) \log pr_{\boldsymbol{\Theta}}(H_l \,|\, X_l = k') \\
&= \sum_{k' \in K'} \log \left[\prod_{l=1}^{L} \{pr_{\boldsymbol{\Theta}}(H_l \,|\, X_l = k')\}^{\Omega_{lk'}(\boldsymbol{\Theta}^{(p)})} \right] \\
&= \sum_{k' \in K'} \log \left[\prod_{i=1}^{s_1} \left[(\rho_i^{(k')})^{w_i^{(k')}} \prod_{k=1}^{M} \{(q_{ik}^{(k')})^{(n_{ik}^{(k')} - d_{ik}^{(k')})} \times \right. \right. \\
&\quad \left. \left. \prod_{j=1}^{s} (\lambda_{ijk}^{(k')})^{m_{ijk}^{(k')}} \} \right] \right], \qquad (2.13)
\end{aligned}$$

where

$$\begin{aligned}
w_i^{(k')} &= \sum_{l=1}^{L} I_{(z_{l,0}=i)} \Omega_{lk'}(\boldsymbol{\Theta}^{(p)}) \\
m_{ijk}^{(k')} &= \sum_{l=1}^{L} m_{ijkl}^{(k')} \Omega_{lk'}(\boldsymbol{\Theta}^{(p)}) \\
d_{ik}^{(k')} &= \sum_{j=1}^{s} m_{ijk}^{(k')} \\
m_{i,s+1,r}^{(k')} &= \sum_{l=1}^{L} I_{(z_{l,m^{(l)}-1}=i, z_{l,m^{(l)}}=s+1, t_{l,m^{(l)}} \in [\nu_r, \nu_{r+1}])} \Omega_{lk'}(\boldsymbol{\Theta}^{(p)}) \\
n_{ik}^{(k')} &= \sum_{r=k}^{M} (d_{ir}^{(k')} + m_{i,s+1,r}^{(k')}).
\end{aligned}$$

It can be seen that the update equations for the mixture weights can be obtained by maximizing (2.12) with respect to the $\pi^{(1)}, ..., \pi^{(K')}$, subject to the constraint $\sum_{k'=1}^{K'} \pi^{k'} = 1$. Solving for the maximum using Lagrange multipliers, we obtain the update equation for the mixture weight as follows (details are given in Appendix A.1):

$$\pi_{new}^{(k')} = \frac{\sum_{l=1}^{L} \Omega_{lk'}(\boldsymbol{\Theta}^{(p)})}{L} + \alpha \pi_{old}^{(k')} (\log \pi_{old}^{(k')} - \sum_{k' \in K'} \pi_{old}^{(k')} \log \pi_{old}^{(k')}). \qquad (2.14)$$

Here, the regularization parameter α is updated using the following relationship:

$$\alpha_{new} = \min\left\{ \frac{e^{(-\kappa|\pi_{new}^{(k')} - \pi_{old}^{(k')}|)}}{K'}, \frac{1 - \max\limits_{k'}(\sum_{l=1}^{L} \Omega_{lk'}(\boldsymbol{\Theta}^{(p)})/L)}{-\max\limits_{k'}\ (\pi_{old}^{(k')}) \sum_{k' \in K'} \pi_{old}^{(k')} \log\ \pi_{old}^{(k')}} \right\}, \tag{2.15}$$

where the constant κ can be set to ηL. The positive constant η can be set to $\min(1, 0.5^{\lfloor m/2-1 \rfloor})$ as in [16], m being the average number of transitions per trajectory. More about this constant is discussed in simulation study.

The update equations for the parameters of the semi-Markov models can be obtained by maximizing (2.13) with respect to the $\{\boldsymbol{\rho}^{(k')},\ \boldsymbol{\lambda}^{(k')}\}_{k'=1}^{K'}$ subject to the appropriate constraints. After working out the details as shown in Appendix A.2, the following update equations are obtained:

$$\rho_{i,npmle}^{(k')} = \frac{w_i^{(k')}}{\sum_{i=1}^{s_1} w_i^{(k')}} \tag{2.16}$$

$$\lambda_{ijk,npmle}^{(k')} = \frac{m_{ijk}^{(k')}}{n_{ik}^{(k')}}. \tag{2.17}$$

Therefore, the conditional probability is obtained as follows:

$$\hat{q}_{ik}^{(k')} = 1 - \sum_{j=1}^{s} \lambda_{ijk,npmle}^{(k')} = \frac{(n_{ik}^{(k')} - d_{ik}^{(k')})}{n_{ik}^{(k')}}. \tag{2.18}$$

Assuming the largest sojourn time in a state i is uncensored, the estimator for $S(t;\, i)$ is given by:

$$\hat{S}^{(k')}(t;\, i) = \hat{q}_{i1}^{(k')} ... \hat{q}_{ik}^{(k')} \ \ (\upsilon_k \leqslant t < \upsilon_{k+1}),$$

for $k = 0, 1, ..., M$; if the largest sojourn time in a state i is censored, $\hat{S}^{(k')}(t;\, i)$ is given by above equation for $t \leqslant \upsilon_M$, but arbitrarily decreases to zero for $t > \upsilon_M$.

Defining $\theta_{ijk}^{(k')} = pr(T_{n+1} = \upsilon_k, Z_{n+1} = j | Z_n = i, X = k')$, then its estimate is

$\hat{\theta}_{ijk}^{(k')} = \lambda_{ijk,npmle}^{(k')} \hat{S}^{(k')}(\upsilon_{k-1};\, i)$. Thus, the estimators of $\theta^{(k')}(i,j)$ and $Q^{(k')}(t;\, i,j)$ are given as follows:

$$\hat{\theta}^{(k')}(i,j) = \sum_{k=1}^{M} \hat{\theta}_{ijk}^{(k')} \tag{2.19}$$

$$\hat{Q}^{(k')}(t;\, i,j) = \frac{\hat{\theta}^{(k')}(i,j) - \sum^{*} \hat{\theta}_{ijk}^{(k')}}{\hat{\theta}^{(k')}(i,j)}, \tag{2.20}$$

where $\sum^{*}$ indicates summation over the set $\{k :\, \upsilon_k \leqslant t\}$. Also, we define the holding time probability as follows:

$$H_{ijk}^{(k')} = pr(T_{n+1} = \upsilon_k | Z_{n+1} = j, Z_n = i, X = k'), \tag{2.21}$$

and its estimate is obtained as follows:

$$\hat{H}_{ijk}^{(k')} = \frac{\lambda_{ijk,npmle}^{(k')} \hat{S}^{(k')}(\upsilon_{k-1};\, i)}{\hat{\theta}^{(k')}(i,j)}. \tag{2.22}$$

Both $\hat{S}^{(k')}(t;\, i)$ and $\hat{Q}^{(k')}(t;\, i,j)$ are the proper distribution functions if the largest sojourn time in a state i is uncensored. However, $\hat{S}^{(k')}(t;\, i)$ is an improper distribution function if the largest sojourn time in a state i is censored. In this case, we assume that it arbitrarily places mass somewhere in the interval (υ_M, ∞). Also, the sum of the transition probabilities $\hat{\theta}^{(k')}(i,1) + ... + \hat{\theta}^{(k')}(i,s)$ is less than one. Therefore, a set of modified estimates can be defined as follows:

$$\tilde{\theta}^{(k')}(i,j) = \frac{\hat{\theta}^{(k')}(i,j)}{\sum_{j=1}^{s} \hat{\theta}^{(k')}(i,j)} \tag{2.23}$$

$$\tilde{Q}^{(k')}(t;\, i,j) = \frac{\tilde{\theta}^{(k')}(i,j) - \sum^{*} \hat{\theta}_{ijk}^{(k')}}{\tilde{\theta}^{(k')}(i,j)}, \tag{2.24}$$

where $\sum^{*}$ indicates summation over the set $\{k :\, \upsilon_k \leqslant t\}$. The clustering and model pa-

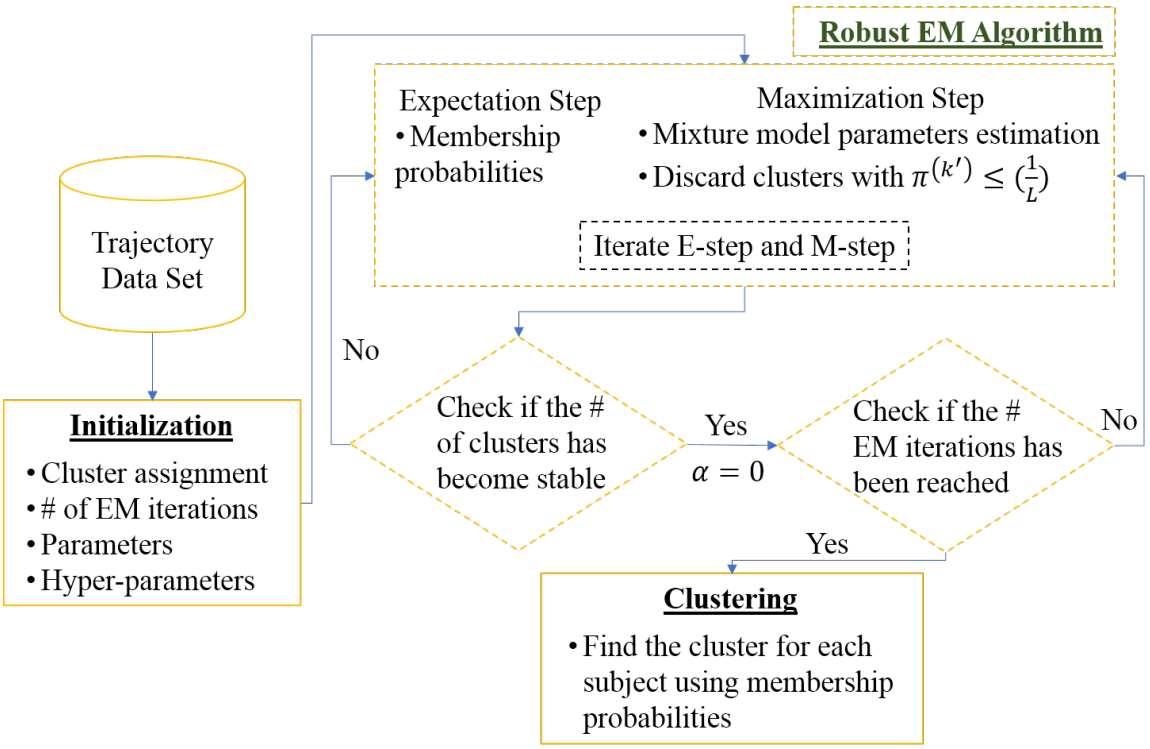

Figure 2.3: Clustering and estimation procedure for trajectory data

rameter estimation framework are summarized in Figure 2.3.

2.5 Simulation Study

In this section, we present the performance of the proposed unsupervised learning approach for clustering censored trajectory. There are two objectives of this simulation study – a) given the number of clusters, assessing the capability of the proposed clustering methodology to perform cluster assignments and mixture model parameters estimation; b) assessing the capability to find the number of clusters as well as perform cluster assignments and mixture model parameters estimation. We simulate data using a mixture of semi-Markov models. Each mixture component follows a different semi-Markov model. For the first objective, we provide the algorithm with the true number of clusters. We check the algorithm's ability to perform cluster assignments and mixture model parameters estimation under different scenarios. In the first scenario, keeping $s_1 = 5$, $K^{'} = 4$, and $L = 9000$,

the censoring fraction is varied from 0 to 1 with a difference of 0.2. We explain what a censoring fraction is by means of an example. Suppose the censoring fraction is 0.4. This means randomly selecting 40% of the total number of trajectories to be right-censored in an uninformative manner. To introduce right censoring in a randomly selected trajectory, we trim it from the end by a randomly selected percentage from 10%, 20%, or 30%. In the second scenario, keeping $K^{'} = 4$, $L = 9000$, and censoring fraction as 0.6, the number of transient states is varied from 2 to 7 with a difference of 1. In the third scenario, keeping $s_1 = 5$, $L = 9000$, and censoring fraction as 0.6, the number of clusters are varied from 2 to 7 with a difference of 1. In the fourth scenario, keeping $s_1 = 5$, $K^{'} = 4$, and censoring fraction as 0.6, the number of trajectories are varied from 5000 to 15000 with a difference of 2000. In all the scenarios, the number of absorbing states is kept fixed as 1.

Using F1-score as the performance metric, Figure 2.4 shows the results for the first objective. A mixture of semi-Markov models without accounting for censoring and a sequence graph transform followed by K-means are the two benchmark methods used for the comparison. As is clearly evident, our approach performs better than benchmarks in all the scenarios. All the methods show nearly decreasing performance with an increase in the censoring fraction because as the number of incomplete trajectories in the data set increases, it becomes harder for the methods to perform accurate clustering. When the number of transient states varies, our method and the mixture of semi-Markov models approach show a quadratic trend. This could be attributed to the fact that a lower number of transient states doesn't lead to the generation of enough variance among the trajectories as required by the fixed number of clusters. After $s_1 = 4$ where performance is the best, we see a decrease in performance as the number of transient states, hence the model complexity, increases. With an increase in the number of clusters, all the methods show a decreasing trend as expected due to the increase in model complexity. Both our method and the mixture of semi-Markov models show an increasing trend with an increase in the number of trajectories as expected, because more data leads to better estimation, hence, clustering. With this, we can conclude

that our proposed clustering algorithm performs better than other existing methods while estimating mixture components knowing the true number of clusters.

The second part of the objective involves checking the proposed methodology's ability to determine the correct number of clusters. In this part, we do not provide the algorithm with the true number of clusters. Instead, we start with a sufficiently large number of clusters, e.g., 15 and let the algorithm decide the optimal number of clusters. Although [16] suggested to start with a number of clusters equal to the number of observations while presenting their algorithm for the case of the mixture of Gaussians, it is impractical in the real scenarios where the number of observations could be of the order of thousands. Still, we recommend starting with a number of clusters as large as possible. Also, in this part of the simulation study, the number of transient states, number of absorbing states, and censoring fraction are kept fixed as 5, 1, and 0.6, respectively. As stated earlier in 2.4, positive constant η is defined as $\min(1, 0.5^{\lfloor m/2-1 \rfloor})$, as suggested in [16] for mixture of Gaussians. Since, in the case of trajectory data, the dimension of each trajectory/observation is not fixed, m needs to be defined as the average number of transitions per trajectory. Also, through our simulation study, we find that this expression for η may not work all the time in the case of the mixture of semi-Markov models with censoring. Therefore, we suggest η can be set as $\min(1, 0.5^{\lfloor m/\tau-1 \rfloor})$, where τ can easily be chosen from $[0.5, 1.5]$. In future work, we plan to explore this further and find a better way to estimate m thereby eliminating the need for τ. Since the robust EM-based algorithm proposed in this work almost resolves the initialization problem associated with the standard EM algorithm and provides an easy way to determine the optimal number of clusters, we would like to emphasize that the proposed algorithm is better than the standard EM algorithm. The proposed algorithm does not require trying different initialization and trying a large number of alternatives for the optimal number of clusters.

Table 2.1 shows the results related to the objective that involved checking the performance of the proposed algorithm in terms of its ability to determine the optimal number

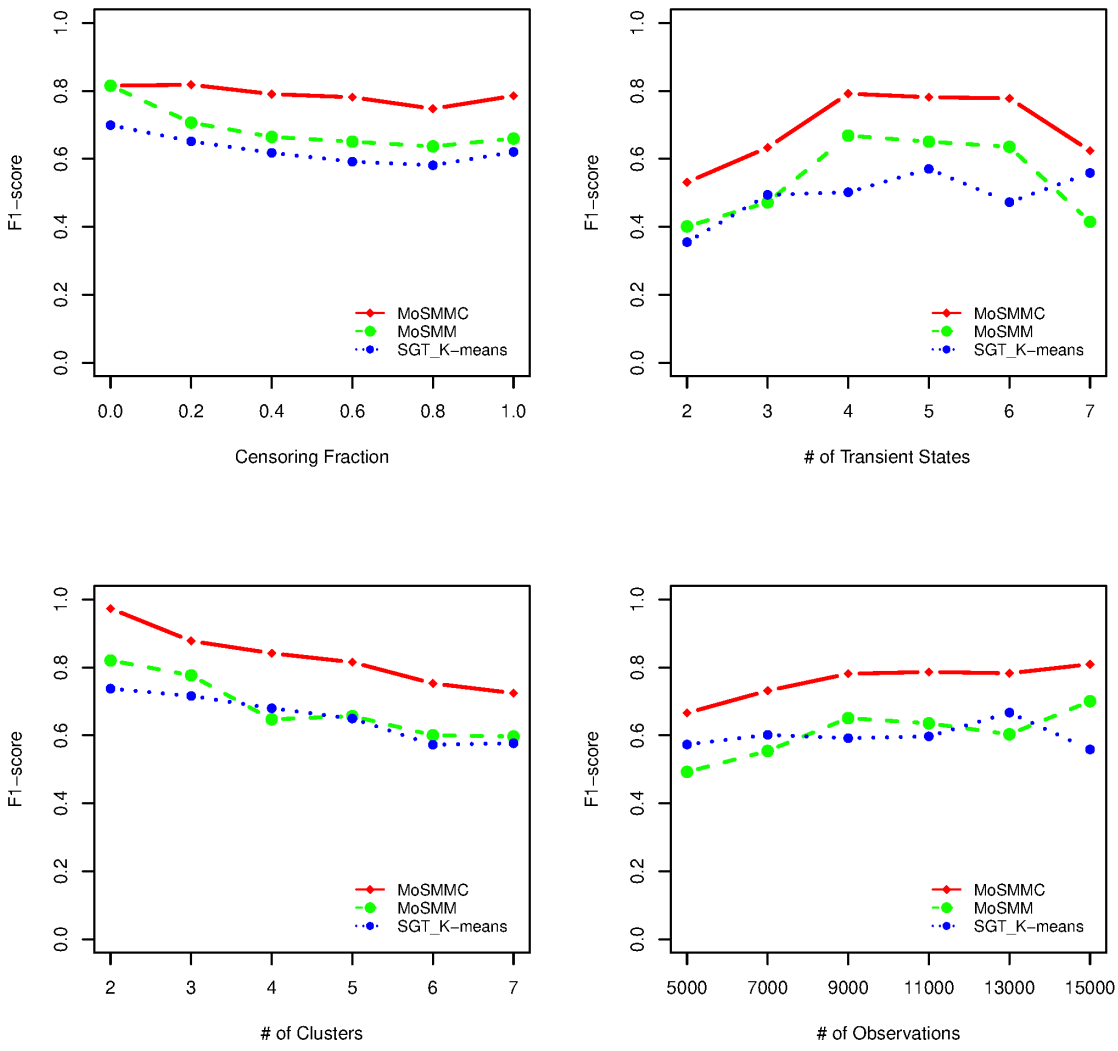

Figure 2.4: Clustering performance comparison under different simulation settings

Table 2.1: Results on proposed clustering algorithm's capability to obtain the optimal number of clusters

# of Trajectories	Hyperparameter (η)	True # of Clusters	Obtained # of Clusters	F1-score
13000	3.051758e-05	3	3	0.9679
15000	3.051758e-05	4	4	0.9163
17000	2.441406e-04	5	5	0.8075
19000	7.629395e-06	6	6	0.8297
21000	7.8125e-03	7	8	0.6239

of clusters. As we can see, our method can determine the number of clusters in almost all of the cases. In the first case, we set $L = 13000$ and simulate data containing 3 mixture components. m is evaluated to be 8 and $\tau = 0.5$ is selected. Thus, η turns out to be $3.051758e-05$. With these settings, the algorithm obtains the number of clusters equal to the true number of clusters with an F1-score of 0.9679. In the second case, we set $L = 15000$ and generate trajectories based on 4 clusters. Parameters m, τ, and η are the same as in the first case. Even in this case, the algorithm determines the correct number of clusters, achieving an F1-score of 0.9163. In the third case, the number of trajectories is set to 17000 and data from 5 mixture components is simulated. Parameter m has the same value as before. $\tau = 0.6$ is chosen. η is calculated as $2.441406e-04$. Again, the optimal number of clusters is found correctly by the algorithm with an F1-score of 0.8075. In the fourth case, the number of observations is 19000 and data consists of 6 clusters. m is found to be 9. $\tau = 0.5$ is selected. η is determined as $7.629395e-06$. Once again, the algorithm finds the correct number of clusters at the same time achieving an F1-score of 0.8297. In the last case, 21000 trajectories are simulated constituting 7 clusters. m is evaluated to be 8. Different values of τ are tried and finally, $\tau = 1$ is chosen, which gives $\eta = 7.8125e-03$. The number of clusters found is 8 which is not the same as true number of clusters but still close enough. Obtained F1-score in this case is 0.6239. With this, we can conclude that our proposed algorithm can determine the optimal number of clusters fairly reliably.

2.6 Case Study

In this section, we use the proposed clustering model and algorithm for a real trajectory data set of internet radio users. A data set consisting of $26,293$ trajectories is obtained. For this data set, the number of transient states is 3 and the number of absorbing states is 1. Three transient states are as follows: free tier denoted as 1, subscription tier denoted as 2, and state of inactivity denoted as 3. All the trajectories are censored. After applying the proposed approach, we obtain 3 clusters. The model parameters associated with each cluster are presented in the form of visualizations as shown in Figures 2.5, 2.6, and 2.7. In each visualization, the size of the node depicts initial-state probability, the width of the edge depicts transition probability, and the edge color or label depicts expected sojourn time.

As it is clear from the size of the nodes of graph visualizations presented in Figures 2.5, 2.6, and 2.7, in cluster # 2, users predominantly start in free tier, whereas in clusters # 1 & # 3, although most of the users start in free tier, some of the users remain idle for some time and don't start listening to radio immediately after registering for the service. From the graph shown in Figure 2.5, we can see that users of cluster # 1 mostly move between the free tier and the state of inactivity, as depicted through the width of the corresponding edges. Also, the very few subscribers who have opted for the service as depicted through small transition probabilities corresponding to the transition from free tier or state of inactivity to subscription tier, tend to move to the free tier, thereby unsubscribing the service. Referring to the graph presented in Figure 2.6 and noticing the edge widths, apart from the majority of the users moving between free tier and state of inactivity, some users of cluster # 2 tend to subscribe to the service. These are relatively more numerous than the subscribers present in cluster # 1. After subscribing to the service, it is more likely that these users will continue service in subscription mode rather than reverting to the free tier. In terms of transition probabilities, the behavior of users in cluster # 3 is similar to those in cluster # 1.

As we have seen in clusters # 1 and # 3, users mostly move between free tier and state of inactivity. It is to be noted that users belonging to cluster # 1 transition to the state of inactivity from the free tier sooner and stays in the state of inactivity for longer than cluster # 3 users, as observed from the edge colors or labels of the graphs shown in Figures 2.5 and 2.7. Also, subscribers in cluster # 3 spend a little more time in subscription tier before they move to the state of inactivity or free tier than users in cluster # 1. From the marketing point of view, it would be better to target cluster #3 users with offers as there is a better chance to minimize these users' tendency to unsubscribe. Cluster # 2 users who are found to be subscribers earlier, can again be seen as active users because they tend to spend more time in subscription tier before they move to either free tier or state of inactivity, as observed from edge colors of the graph shown in Figure 2.6. From an advertisement placement point of view, these users should not be overloaded with advertisements as they are found to be loyal users. Overall, we find the results to be insightful from the marketing and advertisement business point of view.

2.7 Conclusion

Trajectory data is commonly found in different areas such as marketing, reliability, and healthcare. This data is very useful because it promotes understanding of entity (user) behavior. However, there are challenges associated with analyzing such data. First, there is a variety of behaviors present in the data. Second, such data is usually censored. To address these challenges, this work proposes an unsupervised learning framework to cluster a censored trajectory data set. The proposed framework employs a model-based clustering approach. The transition dynamics embedded in the data set is represented by a set of semi-Markov models. Each model also considers the effect of censoring. Parameters of the mixture model are estimated using an EM-based algorithm. Along with the parameters estimation, trajectories are assigned to clusters. The proposed EM-based algorithm can overcome the initialization issue. Also, the algorithm doesn't require the number of clusters

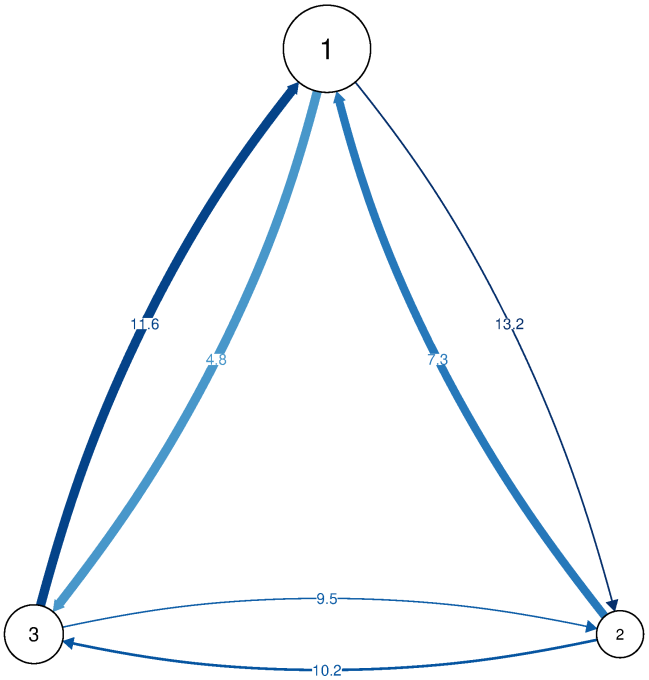

Figure 2.5: Cluster 1 – model parameters summary

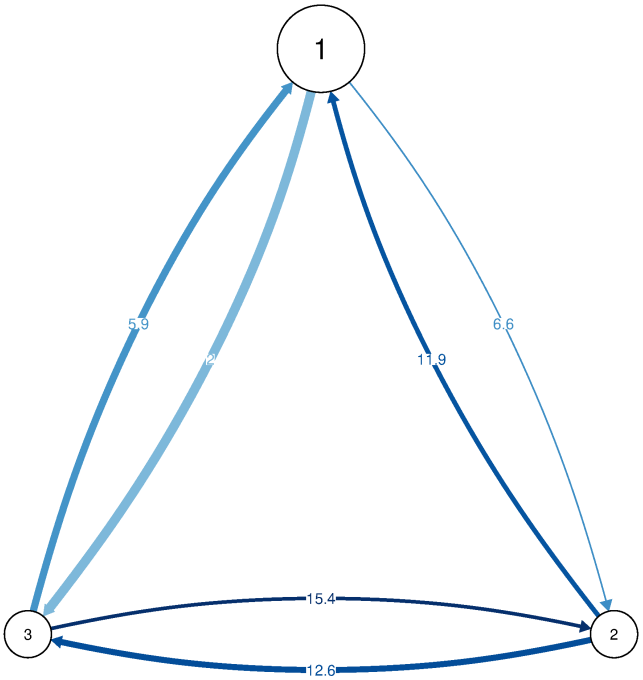

Figure 2.6: Cluster 2 – model parameters summary

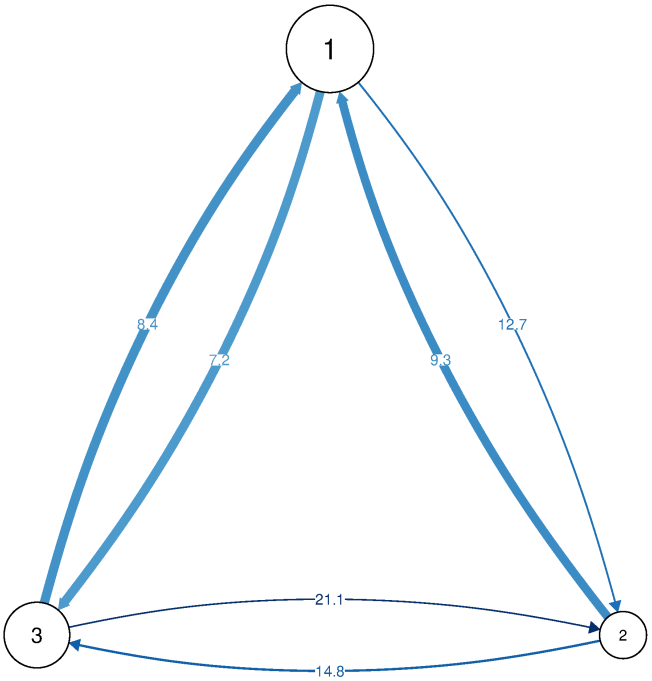

Figure 2.7: Cluster 3 – model parameters summary

to be set a priori because it can determine the optimal number of clusters. Through a simulation study, the effectiveness of the proposed approach is showcased and also compared with other existing methods. The applicability of the proposed framework is demonstrated with the help of a real trajectory data set obtained from an online radio service provider. Effective managerial insights for marketing and advertisement teams are derived. These insights can inform strategies to increase user retention and conversion, and therefore revenue.

CHAPTER 3
IMAGE DECOMPOSITION-BASED SPARSE EXTREME PIXEL-LEVEL FEATURE DETECTION MODEL

3.1 Introduction

Image segmentation is one of the important tasks in computer vision. It involves partitioning images into multiple segments. Each segment can be regarded as consisting of a group of pixel-level features. So, segmentation task can also be regarded as pixel-level feature detection which is encountered in different domains such as manufacturing and medical. For example, (a) in manufacturing systems, an image of a product is sensed for product inspection; (b) in medical domain, an image of a human body part is acquired for disease diagnosis and treatment planning. Both product inspection and disease diagnosis require detecting relevant pixel-level features from the image. In the manufacturing example, feature detection involves extracting defects or anomalies, whereas in the case of medical imaging, it involves segmenting tumors or other biologically relevant structures. Since images are high-dimensional and possess complex spatial structure, feature detection is challenging. Moreover, image sensing and acquisition systems introduce measurement noise that reduces the contrast between the background and different features, thus making the feature detection task even more difficult [26].

In the medical domain, image feature detection is of special importance. Aortic stenosis (AS) is one of the most common yet severe valvular heart diseases. Transcatheter aortic valve replacement (TAVR) is a less-invasive treatment option for AS patients who have a high risk of open-heart surgery. TAVR procedure involves implanting a bioprosthetic aortic valve. Major post-procedural complications of TAVR are the paravalvular leakage (PVL), i.e., the blood flow leakage around the implanted artificial valve [27, 28], and the

over-stretching in the aortic tissues introduced due to the implant. For patients undergoing TAVR, computed tomography (CT) image (see Figure 3.1) is usually taken before the surgery, as an important visualization of the *contrast-enhanced blood pool* (i.e., the moderate intensity region in the CT image), the *calcification* (i.e., the high-intensity region) and the *soft tissues* (i.e., the low-intensity region).

For the clinical decision making and pre-procedural planning, the identification of the soft tissues from the CT image is essential. From the identified soft tissues, the dimensions/structure of the aortic valve can be extracted. Knowledge of the dimensions/structure of the patient's aortic valve helps in selecting an appropriately-sized artificial aortic valve. If the artificial valve is not chosen appropriately, it can trigger post-TAVR complications. For example, if the implanted artificial valve is too small in size, severe PVL may occur, whereas an oversized artificial valve may introduce high stress in the aortic tissues. Another clinically important factor is the calcification present at the aortic annulus region. If the calcification volume and its distribution are effectively quantified directly from the CT image, physicians can better assess the patient's condition and therefore patient-specific treatment planning can be conducted to mitigate the post-TAVR PVL amount.

Furthermore, due to the advancement in 3D-printing technology [29], treatment planning of TAVR can also be assisted by the 3D-printed, patient-specific virtual aortic valve [27, 30]. One of the crucial steps in this approach is to extract a printable digital model (containing aortic valve and calcification) from CT image [31], which again requires precise estimation and detection of the aortic valve region and calcification.

Thus, in order to better plan the TAVR surgery and reduce the post-TAVR complications, a precise estimation of the aortic valve's structure and calcification should be conducted based on the CT image. However, this is a challenging task, mainly due to the complexity involved with the CT data. Currently, the most common practice is to consult radiologists who identify the aortic valve and calcification regions, which can be both time-consuming and sometimes inaccurate. With an aim to reduce human labor, image

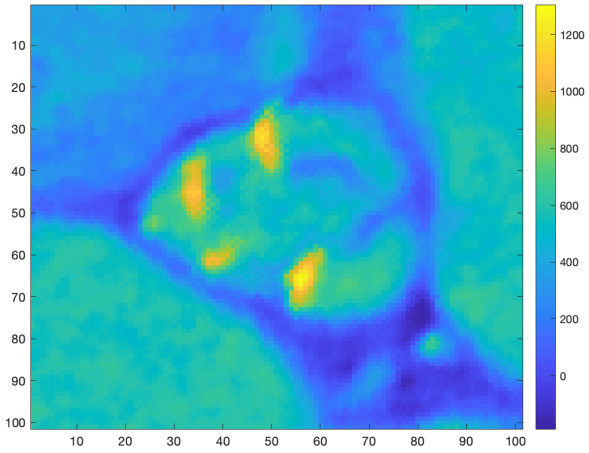

Figure 3.1: An example of a 2-dimensional computed tomographic image, showing the soft tissues (blue), the calcification (bright yellow), and the blood pool (bluish yellow)

segmentation techniques such as global thresholding and multi-level thresholding are also used [32]. Since global thresholding simply performs image binarization using a single threshold, experienced radiologists may be required to extract the desired region from the image. As for multi-level thresholding, several levels of thresholding need to be carried out to partition the image to finally obtain desired regions. However, thresholding-based segmentation approaches do not consider spatial correlation among image pixels. As a result, the selected features may not form connected regions and have pathological meaning. Moreover, due to a huge variation in the pathological condition, different patients often times have completely different contrast agent concentration and distribution of pixel intensity in the CT image. As a result, every patient requires a dedicated analysis.

In this work, we propose a novel, automatic sparse extreme pixel-level feature detection (PFD) model to identify the soft tissues and the calcification. This model decomposes an image into four components – mean, positive extreme features, negative extreme features, and noise. The mean part is assumed to be smooth, containing only the overall tone of the CT image and can be approximated using a basis. The extreme features which refer to both the calcification (i.e., high-intensity region, which is above mean) and the soft

tissues (i.e., low-intensity region, which is below mean), are assumed to be sparse or can be sparsely represented. Their spatial structures can be modeled using different bases. Moreover, appropriate constraints on bases coefficients of features are imposed to keep positive and negative features separate, thereby overcoming the identifiability issues. To estimate the parameters in the PFD model with both sparsity regularization and constraints on bases coefficients, a convex optimization problem is framed, which can be efficiently solved using alternating direction method of multipliers (ADMM) with proximal gradient method (PGM). Using the proposed PFD model, both the calcification and the soft tissues can be identified with high accuracy, thereby permitting the follow-up diagnosis procedure and the treatment planning of TAVR. Moreover, it is worth emphasizing that the PFD model and the associated algorithm can be applicable for other cases as well, where pixel-level feature/anomaly detection from images is required.

The structure of the paper is as follows. Motivation and problem definition have been put forth in this section. The relevant literature is reviewed in Section 2. The proposed modeling and estimation framework is illustrated in Section 3. The simulation study to evaluate the proposed framework is presented in Section 4. In addition, a real data set is used to evaluate the performance of the proposed framework in Section 5. Finally, conclusions are discussed in Section 6.

3.2 Literature Review

Considerable research has been conducted in the area of medical image segmentation. The most commonly used methods are thresholding, edge detection, and region growing. Thresholding methods are further divided into two categories – global and local (multi-level). In global thresholding, an image is divided into two regions: one having pixel intensities less than a threshold and another having intensities greater than that threshold [33]. The threshold can be manually or automatically selected. Using only a single threshold value, finding precise delineation of region boundaries remains a difficult task. In local

(multi-level) thresholding, an image can be segmented into different regions using multiple thresholds [34]. The major drawback of thresholding techniques is that these do not consider the spatial relationships among features present in an image. Due to this, these techniques are very sensitive to noise.

Edge detection methods utilize the gradient information to identify edges in the image. Several edge detection operators such as Sobel, Laplacian, and Canny operators are commonly used [35]. However, edge detection methods are not robust to noise. Moreover, the output of these methods is often discontinuous pixels which may not form a continuous curve or closed region. As a result, most of these methods require a post-processing step.

Region growing methods start with a single seed point or a set of seed pixels and continue exploring the neighboring pixels in a recursive manner. Based on certain criteria such as uniformity and connectivity, it is determined whether the neighboring pixels should be added to the region [36]. However, these methods typically require a set of initial seed pixels, where the (sparse) extreme regions can be grown from.

Active contours, which are also commonly called *snakes*, are used extensively for image segmentation [37]. Snake is an energy-minimizing curve that can move through the spatial domain of an image under the guidance of internal forces designed to hold the curve together and keep it from bending too much, and external forces designed to draw curves toward the desired features [38, 39]. Researchers have developed a semi-automatic segmentation algorithm to extract aortic valve and calcification based on thresholding technique and snake algorithm [32]. Their algorithm involves multiple steps – (a) the Sinus of Valsalva on the multiplanar reconstruction image is manually detected and cropped off; (b) the cropped region is binarized by global thresholding; (c) the manual intervention is again required to place the initial seed points and create an initial contour on which gradient vector flow snake [39] algorithm is applied to extract aortic valve area. To detect calcification, a heuristic-based thresholding approach utilizing a histogram of image pixel intensity is used. The major drawback of this approach is that it requires manual intervention and is

too case-specific.

Atlas-based segmentation techniques have also been used for medical image segmentation. In this type of technique, an atlas or a look-up table containing information on different features is compiled with the help of experts who carry out labeling on a set of existing images [40]. Atlas template and target image are registered so that atlas labels can be propagated to the target image. Deformable subdivision surface fitting is another segmentation technique in which a subdivision surface, a smooth boundary surface controlled by a coarse mesh with local support, is fitted to the image data [41]. Researchers have proposed a multi-step approach combining atlas-based segmentation and deformable subdivision surface fitting to extract aortic root [42]. However, atlas-based segmentation has high computational cost, and initial model in deformable subdivision surface fitting has a great impact on the result.

Statistical shape model (SSM) is another segmentation technique that involves learning the shape of the region of interest using prior knowledge in the form of a training data set [43]. The major limitation of this method is that they require a large number of uniform training samples. Moreover, if the test sample strongly deviates from the training data set, they are prone to failure. In pathological settings, such a scenario is highly likely.

In optimization-based segmentation techniques, the task of image partitioning is formulated as an energy minimization problem where objective function mainly consists of two terms – data term and regularization term. While the data term reflects the strength of association between a pixel and its label, the regularization term includes priors on the space of feasible solutions and deviations from the priors [44, 45]. For more information on optimization-based approaches and other segmentation techniques involving prior knowledge inclusion, readers are referred to [46].

Recently, an image decomposition-based model, known as smooth-sparse decomposition (SSD), has been developed to detect anomalies in noisy images with smooth backgrounds [26]. The SSD model decomposes an image into a smooth mean, sparse anoma-

lies, and noise. It is cast as a penalized high-dimensional regression problem, which is solved using large-scale optimization techniques. Despite being effective in separating anomalies from a smooth background, it doesn't explicitly allow separating positive anomalies/features from negative ones. Post-processing SSD results may help in separating positive features from negative ones, but it is not guaranteed that both types of features can be clearly separated. Since it uses a single basis for anomalies, sometimes desired positive and negative features segregation may not be achieved due to identifiability issues faced by the optimization problem. Although it is possible to extend SSD to a version where different bases for positive and negative anomalies can be allowed to be used, setting different bases for positive and negative features without considering appropriate constraints on the coefficients of the bases may result into identifiability issues and if applied to our case, the resulting features may not have pathological meaning.

Therefore, in this work, we propose an approach which ensures both positive and negative extreme features, approximated by the same basis, are clearly identifiable by imposing constraints on the coefficients of the bases and we also show how to incorporate different bases for positive and negative extreme features to detect them precisely.

3.3 Pixel-level Feature Detection

In this section, we present the proposed PFD model and the optimization algorithm for parameter estimation. The proposed model is flexible in the sense that it can be used to extract either positive or negative extreme features or both from a high-dimensional image. Although in this work we focus on the CT image, the proposed model is generic enough to be applicable for other image types occurring in different domains such as medical imaging and manufacturing systems. Also, the proposed framework is extendable to higher dimensions (e.g., 3-D images). In this study, we are interested in both the positive and negative features because calcification in the CT slice corresponds to those pixel intensities which are much higher in magnitude than the mean intensity and soft tissues/aortic valve

corresponds to lower pixel intensities.

3.3.1 Proposed Model

Let us consider a 2-D image denoted by $Y \in \mathbb{R}^{n_1 \times n_2}$. Since we are interested in detecting the positive and negative features, the following PFD model is proposed to decompose an image into four components:

$$Y = M + A_p + A_n + E, \tag{3.1}$$

where M is the mean, A_p is the positive features, A_n is the negative features, and E is the error. The mean and features can be further expanded using different bases such as B-splines and wavelets. So, the model can be re-written as:

$$Y = B\theta + B_p\theta_p + B_n\theta_n + E, \tag{3.2}$$

where B, B_p, and B_n are the bases for the mean, the positive extreme features, and the negative extreme features, respectively, and θ, θ_p, and θ_n are the corresponding coefficients. Furthermore, we impose constraints in the form of: $\theta_p > 0$ and $\theta_n < 0$. The model presented in equation 3.2 is more flexible than SSD which decomposes an image in the following fashion: $Y = B\theta + B_a\theta_a + E$, where B_a is the basis for anomalies and θ_a are the corresponding coefficients with no explicit constraints on them.

3.3.2 Optimization Algorithm for Parameter Estimation

To estimate the model parameters, θ, θ_p, and θ_n, a least squares regression is used. The least squares regression is augmented with L_1 and L_2 penalties to ensure the sparsity of the detected features and smoothness of the estimated mean, respectively. It is also augmented with constraints on the coefficients of the bases. Consequently, the PFD model parameters

can be estimated by solving the following optimization problem:

$$\begin{aligned} \underset{\theta,\theta_p,\theta_n}{\text{argmin}} \quad & ||\tilde{e}||^2 + \lambda\theta^T R\theta + \gamma_p||\theta_p||_1 + \gamma_n||\theta_n||_1 \\ \text{subject to} \quad & \tilde{y} = B\theta + B_p\theta_p + B_n\theta_n + \tilde{e} \\ & \theta_p > 0 \\ & \theta_n < 0, \end{aligned} \tag{3.3}$$

where $||\cdot||_1$ and $||\cdot||$ are L_1 and L_2 norm operators, respectively; λ, γ_p, and γ_n are parameters to be tuned by the user; R is the roughness matrix; $\tilde{y} = vec(Y)$, $B = B_2 \bigotimes B_1$, $B_p = B_{p,2} \bigotimes B_{p,1}$, $B_n = B_{n,2} \bigotimes B_{n,1}$, $\tilde{e} = vec(E)$, $\bigotimes$ is the tensor product, and $vec(\cdot)$ is an operator that converts a matrix to a column vector. Here, B_i, $B_{p,i}$ and $B_{n,i}$ are the bases along i-th direction ($i = 1$ and $i = 2$ correspond to x and y directions, respectively). It should be noted that the size of the B is determined by its constituting bases - B_2 and B_1. Suppose the size of the B_i is $n_i \times k_{\mu_i}$, in which k_{μ_i} is the number of basis in the i-th direction, then the size of the B is given by $n_1 n_2 \times k_{\mu_1} k_{\mu_2}$. Similarly, suppose that the $B_{p,i}$ is of size $n_i \times k_{p_i}$, then the size of the B_p is $n_1 n_2 \times k_{p_1} k_{p_2}$.

Optimization Algorithm for PFD

To solve the PFD problem in (3.3), we can use general convex optimization solvers like the interior point method [47] since the loss function is convex. However, the interior point method is often slow for high-dimensional data problems and hence cannot be used for cases such as medical images which can turn out to be big in size due to higher resolutions that nowadays can be achieved due to advancements in medical imaging technology. In this section, we propose an efficient algorithm to solve the PFD problem.

First, we derive that the PFD problem in (3.3) can be reduced to a constrained weighted least absolute shrinkage and selection operator (LASSO) [48, 49], as shown in Appendix B.1. The PFD problem is equivalent to a constrained weighted LASSO problem in the form

of:

$$\begin{aligned} \underset{\theta_p,\theta_n}{\text{argmin}} \quad & (\tilde{y} - B_p\theta_p - B_n\theta_n)^T(I-H)(\tilde{y} - B_p\theta_p - B_n\theta_n) + \gamma_p||\theta_p||_1 + \gamma_n||\theta_n||_1 \\ \text{subject to} \quad & \theta_p > 0 \\ & \theta_n < 0, \end{aligned} \tag{3.4}$$

where $H = B(B^TB + \lambda R)^{-1}B^T$.

The constrained weighted LASSO problem in (3.4) can be solved using quadratic programming [49]. However, it may not be efficient in the high-dimensional data setting [26]. Therefore, an efficient algorithm should be developed to solve this problem. We utilize the alternating direction method of multipliers (ADMM) algorithm, which has garnered renewed popularity in statistics and machine learning applications recently [50]. This algorithm is easy to implement and suitable in distributed computing setting as well.

To solve the constrained weighted LASSO problem in (3.4) using the ADMM, it is re-written as follows:

$$\begin{aligned} \underset{\theta_p,\theta_n}{\text{argmin}} \quad & f(\theta_p,\theta_n) + g(z_p,z_n) \\ \text{subject to} \quad & \theta_p - z_p = 0 \\ & \theta_n - z_n = 0, \end{aligned} \tag{3.5}$$

where $f(\theta_p,\theta_n) = (\tilde{y} - B_p\theta_p - B_n\theta_n)^T(I-H)(\tilde{y} - B_p\theta_p - B_n\theta_n) + \gamma_p||\theta_p||_1 + \gamma_n||\theta_n||_1$ and $g(z_p,z_n) = \begin{cases} 0 & \text{if } (z_p,z_n) \in \mathcal{C} \\ \infty & \text{if } (z_p,z_n) \notin \mathcal{C} \end{cases}$, with $\mathcal{C} = \{(z_p,z_n) : z_p > 0, z_n < 0\}$. The functions $f(\theta_p,\theta_n)$ and $g(z_p,z_n)$ need to be closed, proper, and convex, as required by the ADMM. We show in Appendix B.2 that these assumptions are indeed satisfied. Next, the augmented

Lagrangian function is formed as shown below:

$$\begin{aligned}\mathcal{L}_\rho(\theta_p, \theta_n, z_p, z_n, y_p, y_n) \quad &= f(\theta_p, \theta_n) + g(z_p, z_n) + y_p^T(\theta_p - z_p) + y_n^T(\theta_n - z_n) \\ &+ \frac{\rho}{2}(||\theta_p - z_p||^2 + ||\theta_n - z_n||^2),\end{aligned} \tag{3.6}$$

where $\rho > 0$. The ADMM algorithm employs block coordinate descent to the augmented Lagrangian followed by an update of the dual variables as shown below:

$$\begin{aligned}(\theta_p^{(k)}, \theta_n^{(k)}) \quad &\leftarrow \underset{\theta_p, \theta_n}{\text{argmin}}\ \mathcal{L}_\rho(\theta_p, \theta_n, z_p^{(k-1)}, z_n^{(k-1)}, y_p^{(k-1)}, y_n^{(k-1)}) \\ (z_p^{(k)}, z_n^{(k)}) \quad &\leftarrow \underset{z_p, z_n}{\text{argmin}}\ \mathcal{L}_\rho(\theta_p^{(k)}, \theta_n^{(k)}, z_p, z_n, y_p^{(k-1)}, y_n^{(k-1)}) \\ y_p^{(k)} \quad &\leftarrow y_p^{(k-1)} + \rho(\theta_p^{(k)} - z_p^{(k)}) \\ y_n^{(k)} \quad &\leftarrow y_n^{(k-1)} + \rho(\theta_n^{(k)} - z_n^{(k)}),\end{aligned} \tag{3.7}$$

where super-indices (k) and $(k-1)$ denote iteration numbers. Using (3.6) and letting $u_p = y_p/\rho$ & $u_n = y_n/\rho$, the updates in (3.7) can be re-written as follows:

$$\begin{aligned}(\theta_p^{(k)}, \theta_n^{(k)}) \quad &\leftarrow \underset{\theta_p, \theta_n}{\text{argmin}}\ f(\theta_p, \theta_n) + \frac{\rho}{2}(||\theta_p - z_p^{(k-1)} + u_p^{(k-1)}||^2 + ||\theta_n - z_n^{(k-1)} + u_n^{(k-1)}||^2) \\ (z_p^{(k)}, z_n^{(k)}) \quad &\leftarrow \underset{z_p, z_n}{\text{argmin}}\ g(z_p, z_n) + \frac{\rho}{2}(||\theta_p^{(k)} - z_p + u_p^{(k-1)}||^2 + ||\theta_n^{(k)} - z_n + u_n^{(k-1)}||^2) \\ u_p^{(k)} \quad &\leftarrow u_p^{(k-1)} + \theta_p^{(k)} - z_p^{(k)} \\ u_n^{(k)} \quad &\leftarrow u_n^{(k-1)} + \theta_n^{(k)} - z_n^{(k)}.\end{aligned} \tag{3.8}$$

Next, we find the solutions for $(\theta_p^{(k)}, \theta_n^{(k)})$ and $(z_p^{(k)}, z_n^{(k)})$ update problems. First, let's take up $(\theta_p^{(k)}, \theta_n^{(k)})$ update problem. Substituting for $f(\theta_p, \theta_n)$ and re-arranging, it turns out

that the following weighted LASSO problem needs to be solved:

$$\begin{aligned}\underset{\theta_p,\theta_n}{\text{argmin}} \quad & (\tilde{y} - B_p\theta_p - B_n\theta_n)^T(I-H)(\tilde{y} - B_p\theta_p - B_n\theta_n) \\ & + \frac{\rho}{2}(||\theta_p - z_p^{(k-1)} + u_p^{(k-1)}||^2 + ||\theta_n - z_n^{(k-1)} + u_n^{(k-1)}||^2) \\ & + \gamma_p||\theta_p||_1 + \gamma_n||\theta_n||_1.\end{aligned} \tag{3.9}$$

The weighted LASSO problem in (3.9) is first solved for θ_p while keeping $\theta_n = \theta_n^{(k-1)}$. Therefore, the weighted LASSO problem in (3.9) simplifies into following:

$$\begin{aligned}\underset{\theta_p}{\text{argmin}} \quad & (\tilde{y} - B_p\theta_p - B_n\theta_n^{(k-1)})^T(I-H)(\tilde{y} - B_p\theta_p - B_n\theta_n^{(k-1)}) \\ & + \frac{\rho}{2}||\theta_p - z_p^{(k-1)} + u_p^{(k-1)}||^2 \\ & + \gamma_p||\theta_p||_1.\end{aligned} \tag{3.10}$$

The LASSO in 3.10 can be solved using least angle regression (LARS) [51] and quadratic programming [52]. However, these methods are inefficient in the high-dimensional setting [26]. Therefore, an efficient algorithm based on the proximal gradient (PG) method [53, 54] is developed.

The PG method is a popular optimization algorithm which can be used to solve a class of optimization problems that involves sum of a group of convex functions, some of which can be non-differentiable, in the objective function. In our case, the problem in (3.10) has an objective function which consists of $F(\theta_p) = (\tilde{y} - B_p\theta_p - B_n\theta_n^{(k-1)})^T(I-H)(\tilde{y} - B_p\theta_p - B_n\theta_n^{(k-1)}) + \frac{\rho}{2}||\theta_p - z_p^{(k-1)} + u_p^{(k-1)}||^2$ and $G(\theta_p) = \gamma_p||\theta_p||_1$. First term of $F(\theta_p)$ is already proven to be convex in B.2. Second term of $F(\theta_p)$ is also convex. Since $F(\theta_p)$ is a weighted sum of two convex functions, it is also convex. It is easy to see that both the terms in $F(\theta_p)$ are differentiable as well. So, $F(\theta_p)$ is convex-differentiable. Since $G(\theta_p)$ is an L_1 norm, it is convex but non-differentiable. The PG method also assumes that the convex differentiable function $F(\theta_p)$ has an L-Lipschitz continuous gradient. We prove in Proposition 1 that $F(\theta_p)$ is indeed Lipschitz continuous, which can guarantee the

convergence of PG method.

Proposition 1 *$F(\cdot)$ is Lipschitz continuous, i.e., there is a constant $L = 2||B_p||_2^2 + \rho$, where $||\cdot||_2^2$ represents square of matrix spectral norm, which ensures gradient $\nabla F(\cdot)$ satisfy $||\nabla F(\alpha) - \nabla F(\beta)|| \leq L||\alpha - \beta||$, $\forall\, \alpha, \beta \in \mathbb{R}$.*

Proof of the Proposition 1 can be found in Appendix B.3.

Consequently, the PG method optimizes the weighted LASSO problem in 3.10 via an iterative algorithm given by:

$$\underset{\theta_p}{\text{argmin}} \quad F(\theta_p^{(k-1)}) + \langle \theta_p - \theta_p^{(k-1)}, \nabla F(\theta_p^{(k-1)}) \rangle + \frac{L}{2}||\theta_p - \theta_p^{(k-1)}||^2 + \gamma_p||\theta_p||_1, \quad (3.11)$$

where super-indices (k) and $(k-1)$ denote iteration numbers and $\langle \cdot, \cdot \rangle$ refers to the inner product operator. We prove in Proposition 2 that the PG algorithm has a closed-form solution in each iteration k.

Proposition 2 *The proximal gradient algorithm for the problem in 3.10, given by $\theta_p^{(k)} = \underset{\theta_p}{argmin}\, \{F(\theta_p^{(k-1)}) + \langle \theta_p - \theta_p^{(k-1)}, \nabla F(\theta_p^{(k-1)}) \rangle + \frac{L}{2}||\theta_p - \theta_p^{(k-1)}||^2 + \gamma_p||\theta_p||_1\}$, has a closed-form solution in each iteration k, in the form of a soft-thresholding function as given below:*

$$\theta_p^{(k)} = S_{\frac{\gamma_p}{L}}(\theta_p^{(k-1)} + \frac{2}{L}B_p^T(\tilde{y} - B\theta^{(t-1)} - B_p\theta_p^{(k-1)} - B_n\theta_n^{(k-1)}) - \rho(\theta_p^{(k-1)} - z_p^{(k-1)} + u_p^{(k-1)})). \quad (3.12)$$

Proof of the Proposition 2 is available in B.4.

Similarly, θ_n can be updated using the following:

$$\theta_n^{(k)} = S_{\frac{\gamma_n}{L}}(\theta_n^{(k-1)} + \frac{2}{L}B_n^T(\tilde{y} - B\theta^{(t-1)} - B_p\theta_p^{(k-1)} - B_n\theta_n^{(k-1)}) - \rho(\theta_n^{(k-1)} - z_n^{(k-1)} + u_n^{(k-1)})). \quad (3.13)$$

Now, a solution to $(z_p^{(k)}, z_n^{(k)})$ update problem is discussed. The following problem

needs to be solved:

$$\underset{z_p, z_n}{\text{argmin}} \quad g(z_p, z_n) + \frac{\rho}{2}(||\theta_p^{(k)} - z_p + u_p^{(k-1)}||^2 + ||\theta_n^{(k)} - z_n + u_n^{(k-1)}||^2). \tag{3.14}$$

It is easy to see that the solution to the problem in 3.14 is given by:

$$(z_p^{(k)}, z_n^{(k)}) \quad \leftarrow \text{proj}_{\mathcal{C}}(\theta_p^{(k)} + u_p^{(k-1)}, \theta_n^{(k)} + u_n^{(k-1)}) \tag{3.15}$$

To compute the projection matrix, $H = B(B^TB + \lambda R)^{-1}B^T$, a matrix inversion operation is required, which can be computationally expensive as the size of the image increases. Following [55, 26], the matrix R is defined as $R = B_2^TB_2 \bigotimes D_1^TD_1 + D_2^TD_2 \bigotimes B_1^TB_1 + \lambda D_2^TD_2 \bigotimes D_1^TD_1$, where D_i is the first order difference matrix in the $i-th$ direction. This results in a decomposable projection matrix, $H = H_2 \bigotimes H_1$, where $H_i = B_i(B_i^TB_i + \lambda R)^{-1}B_i^T$. With this, the algorithm becomes efficient for images because the matrix inversion operator is run on lower-dimensional matrices. Hence, the computational complexity of the matrix inversion operation is reduced from $O((k_{\mu_1}k_{\mu_2})^3)$ to $O(k_{\mu_1}^3 + k_{\mu_2}^3)$.

Tuning Parameters Selection - λ, γ_p, γ_n, ρ

Choosing the right values of hyper-parameters is important in optimization. There are three tuning parameters, λ, γ_p, and γ_n, associated with the PFD model. The other tuning parameter ρ is associated with the ADMM algorithm.

The optimal λ is selected based on the generalized cross-validation (GCV) criterion [56] as given below:

$$\lambda_{optimal} = \underset{\lambda}{\text{argmin}}\ \text{GCV}(\lambda) = \frac{||Y - H_1(\lambda)(Y - A_p - A_n)H_2(\lambda) - A_p - A_n||^2/n}{(1 - n^{-1}tr(H(\lambda)))^2}, \tag{3.16}$$

where $A_p = B_{p,1}X_pB_{p,2}^T$ and $A_n = B_{n,1}X_nB_{n,2}^T$. The $H_i(\lambda)$ involves matrix inversion and it needs to be calculated for various values of λ. This can be computationally expensive.

Following [26, 57], a series of transformations and operations are used to speed up the computations. First, Cholesky decomposition of $B_i^T B_i$ is calculated giving square matrix Z_i. Next, the eigenvalues and eigenvectors of matrix $Z_i^{-1} D_i^T D_i Z_i^{-1}$ is calculated as follows: $Z_i^{-1} D_i^T D_i Z_i^{-1} = U_i \text{diag}(s_i) U_i^T$. Computing $V_i = B_i Z_i^{-1} U_i$ prior to optimization, then in each iteration, $H_i(\lambda)$ can be calculated using $H_i(\lambda) = V_i (I + \lambda \text{diag}(s_i))^{-1} V_i^T$ and $tr(H_i(\lambda)) = \sum_{j=1}^{n} \frac{1}{(1+\lambda s_{ij})}$ [26]. It can be easily spotted that now computing $H_i(\lambda)$ doesn't involve matrix inversion and the trick makes its calculation much more efficient.

Instead of choosing optimal γ_p, γ_n, and ρ using a cross-validation method, we adopt an iterative procedure which dynamically tunes these parameters in each iteration of the ADMM algorithm. The optimal γ_p and γ_n can also be selected using the GCV criterion. But, the GCV leads to a higher false positive rate by selecting more pixels because it can be seen from (3.16) that the GCV tries to minimize a residual sum of squares (RSS). In anomaly detection/feature detection, the objective is to extract the anomalous regions/features and not to achieve a smaller RSS [26]. Hence, the Otsu's method [33] is utilized to select γ_p^k & γ_n^k individually. Here, the procedure is explained for selecting γ_p^k and similar procedure applies to γ_n^k as well. Assume that we are interested in segregating x into two classes - background ($j = 1$) and positive anomalous region ($j = 2$). Suppose, weights $\omega_j(\tau)$ denote probabilities of the two classes separated by a threshold τ; $\sigma_j^2(\tau)$ denote the variances of these classes; μ_j denote class means; $p(\cdot)$ denote the histogram of x. Define, $\omega_1(\tau) = \sum_{l=0}^{\tau} p(l)x(l)$, $\omega_2(\tau) = 1 - \omega_1(\tau)$, $\mu_1(\tau) = [\sum_{l=0}^{\tau} p(l)x(l)]/\omega_1$, $\mu_2(\tau) = [\sum_{l=\tau+1}^{L} p(l)x(l)]/\omega_1$. As per Otsu's method, minimizing the intra-class variance $\sigma_w^2(\tau) = \omega_1(\tau)\sigma_1^2(\tau) + \omega_2(\tau)\sigma_2^2(\tau)$ and maximizing the interclass variance $\sigma_b^2(\tau) = \sigma^2 - \sigma_w^2(\tau) = \omega_1(\tau)\omega_2(\tau)[\mu_1(\tau) - \mu_2(\tau)]^2$ are the same. Therefore, by using Otsu's method i.e. maximizing the interclass variance, the optimal threshold used to segregate the classes can be obtained. In our algorithm, we utilize the notation $otsu(x)$ to represent the function which outputs $\gamma^{(k)}$ by applying Otsu's method on x.

As far as ρ is concerned, it could be set at a fixed value pre-determined by simulation

experiments or based on experience. Another option is to use different parameters $\rho^{(k)}$ for each iteration. Let primal residual be defined as $r_k = \sqrt{||\Theta_p^{(k)} - Z_p^{(k)}||^2 + ||\Theta_n^{(k)} - Z_n^{(k)}||^2}$ and dual residual as $s_k = \sqrt{||Z_p^{(k)} - Z_p^{(k-1)}||^2 + ||Z_n^{(k)} - Z_n^{(k-1)}||^2}$. The parameter update scheme used is as follows:

$$\rho^{(k)} = \begin{cases} \kappa^{incr}\rho^{(k-1)} & \text{if } r^k > \mu s^k \\ \rho^{(k-1)}/\kappa^{decr} & \text{if } r^k < \mu^{-1} s^k \\ \rho^{(k)} & \text{otherwise,} \end{cases} \tag{3.17}$$

where $\mu > 1$, $\kappa^{incr} > 1$, and $\kappa^{decr} > 1$ are parameters. We use $\mu = 10$ and $\kappa^{incr} = \kappa^{decr} = 1.025$. This scheme helps in keeping primal and dual residuals close to one another as they both converge to zero. It is to be noted that the ADMM convergence theory is applicable only when ρ is kept constant. So, when using the aforementioned scheme, ρ should be made fixed after a certain number of iterations.

To solve the PFD problem, the detailed optimization algorithm is given in Algorithm 1. Since the optimization problem is convex, from the property of the ADMM and the PG method, it can easily be shown that the proposed algorithm is able to converge to the global optimum [50, 54].

Algorithm 1 Optimization algorithm for solving PFD

Initialization:

Select the bases for background as B_1, B_2, positive features as $B_{p,1}, B_{p,2}$, and negative features as $B_{n,1}, B_{n,2}$,

$Z_i = (B_i^T B_i)^{1/2}$, $i = 1, 2$

$U_i \text{diag}(s_i) U_i^T = Z_i^{-1} D_i^T D_i Z_i^{-1}$, $i = 1, 2$

$V_i = B_i Z_i^{-1} U_i$, $i = 1, 2$

$H_i(\lambda) = V_i(I + \lambda diag(s_i))^{-1} V_i^T$, $i = 1, 2$

$L_p = 2||B_p||_2^2 + \rho$, $L_n = 2||B_n||_2^2 + \rho$

$X_p^{(0)} = 0$, $X_n^{(0)} = 0$, $t = 1$, $\epsilon = 10^{-6}$

while $||X_p^{(t)} - X_p^{(t-1)}|| + ||X_n^{(t)} - X_n^{(t-1)}|| > \epsilon$ **do**

$A_p^{(t-1)} = B_{p,1} X_p^{(t-1)} B_{p,2}^T$; $A_n^{(t-1)} = B_{n,1} X_n^{(t-1)} B_{n,2}^T$

$M^{(t)} = H_1(Y - A_p^{(t-1)} - A_n^{(t-1)}) H_2$

$\Theta_p^{(0)} = X_p^{(t-1)}$, $\Theta_n^{(0)} = X_n^{(t-1)}$, $Z_p^{(0)} = 0$, $Z_n^{(0)} = 0$, $U_p^{(0)} = 0$, $U_n^{(0)} = 0$, $k = 1$

while $||\Theta_p^{(k-1)} - Z_p^{(k-1)}|| + ||\Theta_n^{(k-1)} - Z_n^{(k-1)}|| > \epsilon$ ***OR*** $||Z_p^{(k)} - Z_p^{(k-1)}|| + ||Z_n^{(k)} - Z_n^{(k-1)}|| > \epsilon$ **do**

$A_p^{(k-1)} = B_{p,1} \Theta_p^{(k-1)} B_{p,2}^T$; $A_n^{(k-1)} = B_{n,1} \Theta_n^{(k-1)} B_{n,2}^T$

$\Theta_{pe}^{(k)} = \Theta_p^{(k-1)} + \frac{2}{L_p} B_{p,1}^T (Y - M^{(k)} - A_p^{(k-1)} - A_n^{(k-1)}) B_{p,2} - \rho(\Theta_p^{(k-1)} - Z_p^{(k)} + U_p^{(k)})$

$\Theta_{ne}^{(k)} = \Theta_n^{(k-1)} + \frac{2}{L_n} B_{n,1}^T (Y - M^{(k)} - A_p^{(k-1)} - A_n^{(k-1)}) B_{n,2} - \rho(\Theta_n^{(k-1)} - Z_n^{(k)} + U_n^{(k)})$

$\gamma_p / L_p = Otsu(\Theta_{pe}^{(k)})$; $\gamma_n / L_n = Otsu(\Theta_{ne}^{(k)})$

$\Theta_p^{(k)} = S_{\gamma_p / L_p}(\Theta_{pe}^{(k)})$; $\Theta_n^{(k)} = S_{\gamma_n / L_n}(\Theta_{ne}^{(k)})$

$Z_p^{(k)} = \text{proj}_{\mathcal{C}}(\Theta_p^{(k)})$; $Z_n^k = \text{proj}_{\mathcal{C}}(\Theta_n^{(k)})$

$U_p^{(k)} = U_p^{(k-1)} + \Theta_p^{(k)} - Z_p^{(k)}$; $U_n^{(k)} = U_n^{(k-1)} + \Theta_n^{(k)} - Z_n^{(k)}$

end

end

Basis Selection

Selecting the appropriate bases for the background and the features is also important in the implementation of the PFD model. For the background, depending on whether it's smooth or non-smooth, a selection can be made from a variety of bases. If the background is smooth, a spline or kernel basis can be considered. Hyper-parameters associated with the splines (e.g, number of knots) and kernels (e.g., scale parameter) can be tuned using some criterion such as GCV [26]. If the background is non-smooth, wavelet family can be utilized, and appropriate wavelet type and level of decomposition can be chosen depending on a case by case basis. For example, researchers have used a wavelet basis for representing the underlying cardiac image without tag patterns [58].

As far as features are concerned, having some prior information on the shape of features is preferable. Some ideas for selecting basis are as follows: (a) if the features are in the form of small regions scattered over the background or are thin lines, then one can use identity basis, $B_a = I$; (b) if the features are regions with sharp corners, then linear B-splines is recommended; (c) if the features are regions with curved boundaries, higher order splines such as quadratic and cubic, can be utilized [26, 59]. Also, knowing an estimate of the size of the feature can help in selecting the optimal number of knots for a spline basis. For representing textures, a discrete cosine basis can be utilized [58].

3.4 Simulation Study

In this section, the usability of the proposed PFD model and estimation procedure for feature detection in image data is showcased. The main objective here is to demonstrate – (a) how the proposed model overcomes identifiability issues as often faced by SSD, one of the existing methods, (b) the flexibility of the proposed model in terms of allowing different bases to be used for positive and negative extreme regions. The performance of the proposed PFD model and its comparison with the SSD is evaluated using data simulated under

different conditions. Two other benchmarks, Nick local thresholding and global thresholding, are also used for comparison. Two types of scenarios are considered – (a) both positive and negative features are of the same type, (b) positive and negative features are of different types.

Images are simulated using the following model $Y = M + A_p + A_n + E$, where Y is the simulated image, M denotes true mean, A_p and A_n are positive and negative features, respectively, and E is the error. Size of each simulated image is 128×128. In the present case, mean is simulated using $M(x, y) = exp(-\frac{x^2+y^2}{4})$, where $x = \frac{i}{128}, y = \frac{j}{128}, i = j = 1, ..., 128$. To simulate random noise, E is sampled from i.i.d $N(0, \phi^2)$ with $\phi = 0.05$.

For both the methods – SSD and PFD, the cubic B-spline with 3×3 knots is used for the mean estimation. Since the focus here is to show the PFD method in action when the SSD method faces identifiability issues and is not capable of allowing different bases for features, features are simulated using known bases.

The following performance metrics are used to compare the performance of the methods – background recovery square root mean square error (e_M) defined as the square root of mean square error of the mean estimator $\hat{M}$: $e_M = \sqrt{||M - \hat{M}||^2}$; positive features recovery square root mean square error (e_{A_p}) of the positive features estimator $\hat{A}_p$: $e_{A_p} = \sqrt{||A_p - \hat{A}_p||^2}$; negative features recovery square root mean square error (e_{A_n}) of the negative features estimator $\hat{A}_n$: $e_{A_n} = \sqrt{||A_n - \hat{A}_n||^2}$; features recovery square root mean square error: $e_A = \sqrt{||A_n - \hat{A}_n||^2 + ||A_p - \hat{A}_p||^2}$; precision defined as the percentage of recovered features by the algorithm that are indeed features; and recall defined as the percentage of the true features detected by the algorithm.

3.4.1 Scenario A - same type of positive and negative features

In this scenario, both positive and negative features are simulated using a same type of basis B_a, which is chosen as cubic B-spline with 12×12 knots. Specifically, positive features are obtained using $A_p = B_a A_{ps} B_a^T$, where A_{ps} is a sparse matrix with size 13×13 in which two

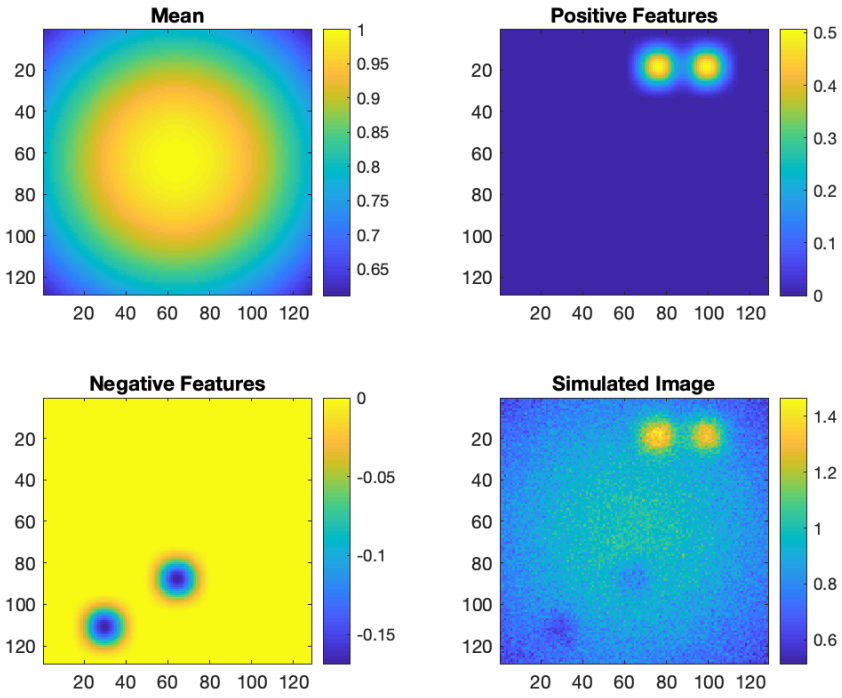

Figure 3.2: Scenario 1 – a sample simulated image with $\delta_p = 0.9$ and $\delta_n = -0.3$

randomly selected entries are replaced with a small number $\delta_p > 0$. Negative features are obtained using $A_n = B_n A_{ns} B_n^T$, where A_{ns} is a sparse matrix of size 13×13 in which two-randomly selected entries are replaced with a small number $\delta_n < 0$. A sample simulated image along with mean, positive features, and negative features is shown in Figure 3.2.

In the experiments discussed in this scenario, we fix and use $\delta_n = -0.3$. We then try different values for δ_p. The recovery square root mean square error performance metrics for SSD with post-processing (PP) (to separate positive and negative features), and PFD are compared in Figures 3.3 and 3.4. As it can be seen from Figure 3.3, e_M for the PFD remains almost constant with an increase in δ_p, whereas that for the SSD (PP) becomes higher starting with $\delta_p = 0.9$. The performance metric, e_{A_p}, remains almost constant and same for both the methods as δ_p increases. The metric, e_{A_n}, for PFD remains very low with an increase in δ_p. On the other hand, there is a sudden jump in e_{A_n} for SSD (PP) at and after $\delta_p = 0.9$. This indicates the onset of the identifiability issue here. When feature recovery

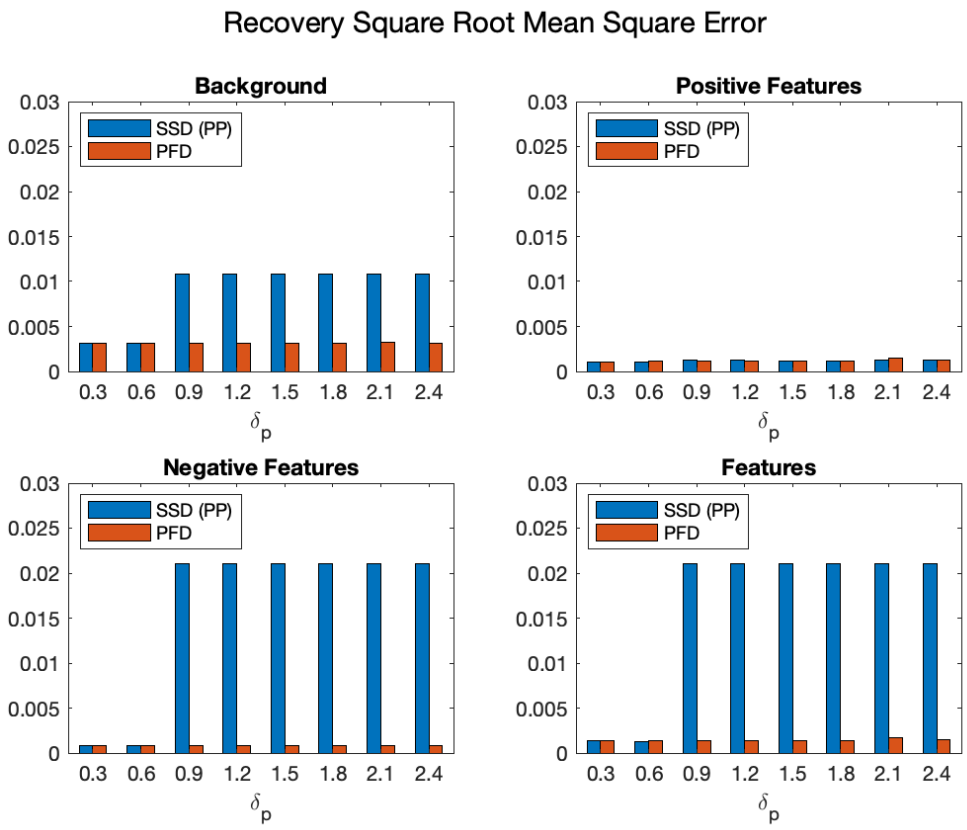

Figure 3.3: Scenario 1 – recovery square root mean square error results showing comparison between SSD (PP) and PFD

performance is seen through a combined metric, e_A, PFD clearly outperforms SSD (PP). Next, we discuss the performance in terms of precision and recall as illustrated in Figure 3.4. PFD maintains perfect precision and recall scores when δ_p is varied, whereas SSD (PP) is unable to detect negative features starting with $\delta_p = 0.9$ because it doesn't allow selecting separate thresholds for positive and negative features. This phenomenon is illustrated with the help of Figures 3.5 and 3.6. Although Nick local thresholding maintains a perfect recall, it's precision is very low in comparison to PFD. Global thresholding shows poor performance with very low precision and recall among all the methods. The sample results for the Nick local and the global thresholding are shown in Figure 3.7. In this scenario, the PFD clearly outperforms the existing methods in terms of all the performance metrics.

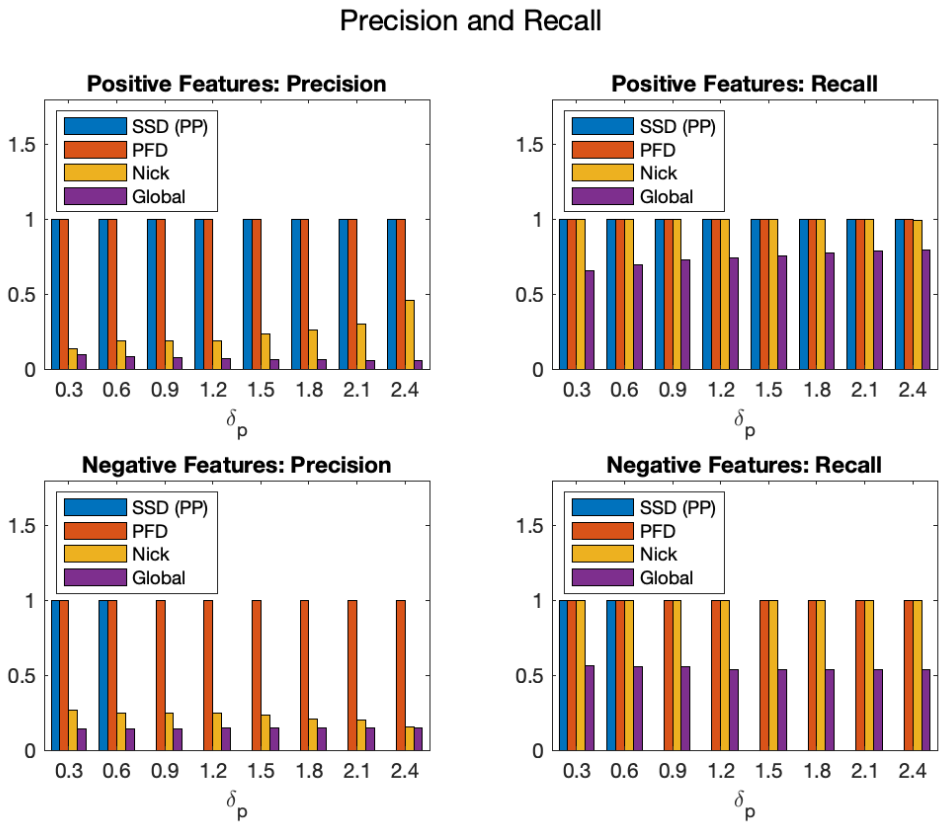

Figure 3.4: Scenario 1 – precision and recall results showing comparison among SSD (PP), PFD, Nick local thresholding, and global thresholding

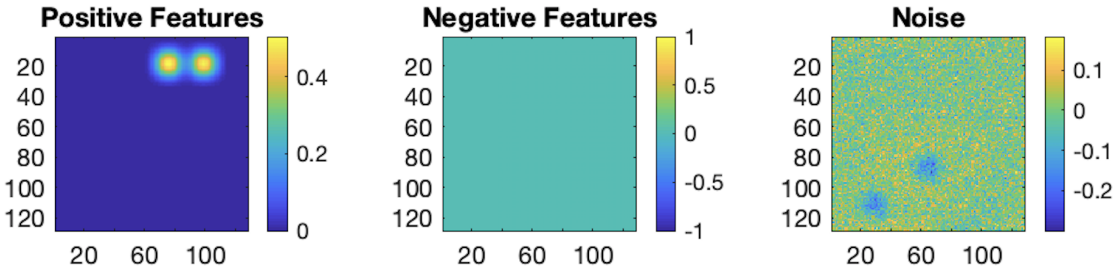

Figure 3.5: Scenario 1 – identifiability issue faced by SSD (PP), when $\delta_p = 0.9$

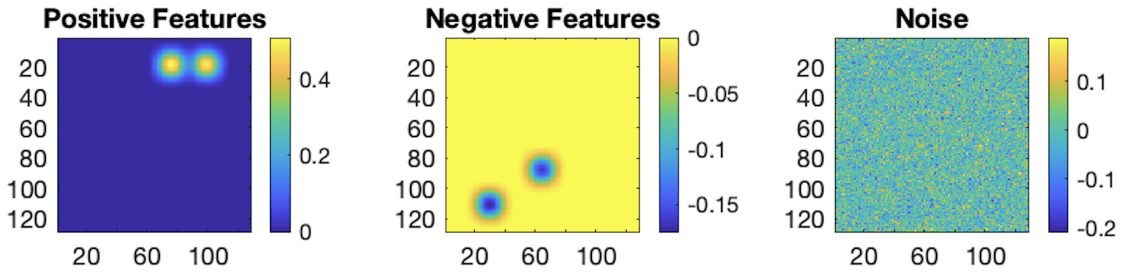

Figure 3.6: Scenario 1 – identifiability successfully dealt with PFD, when $\delta_p = 0.9$

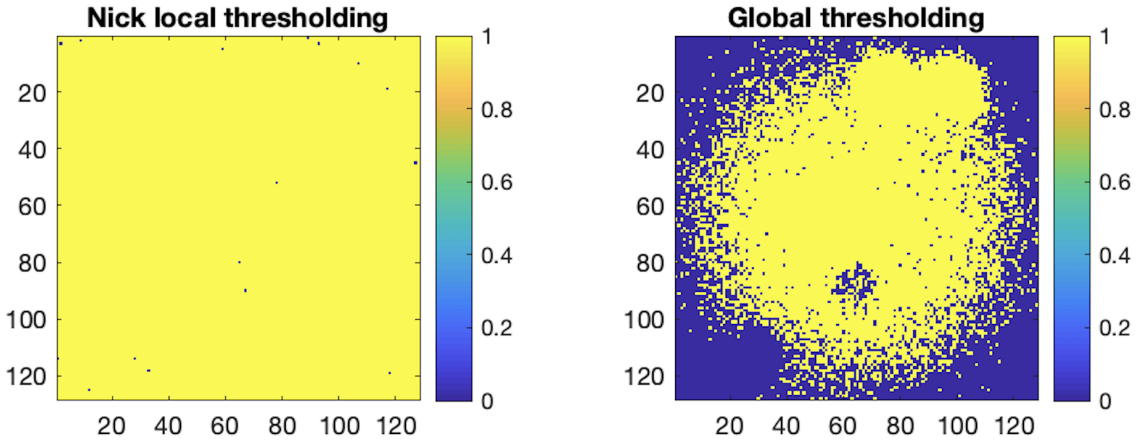

Figure 3.7: Scenario 1 – features detected by Nick local thresholding and global thresholding, when $\delta_p = 0.9$

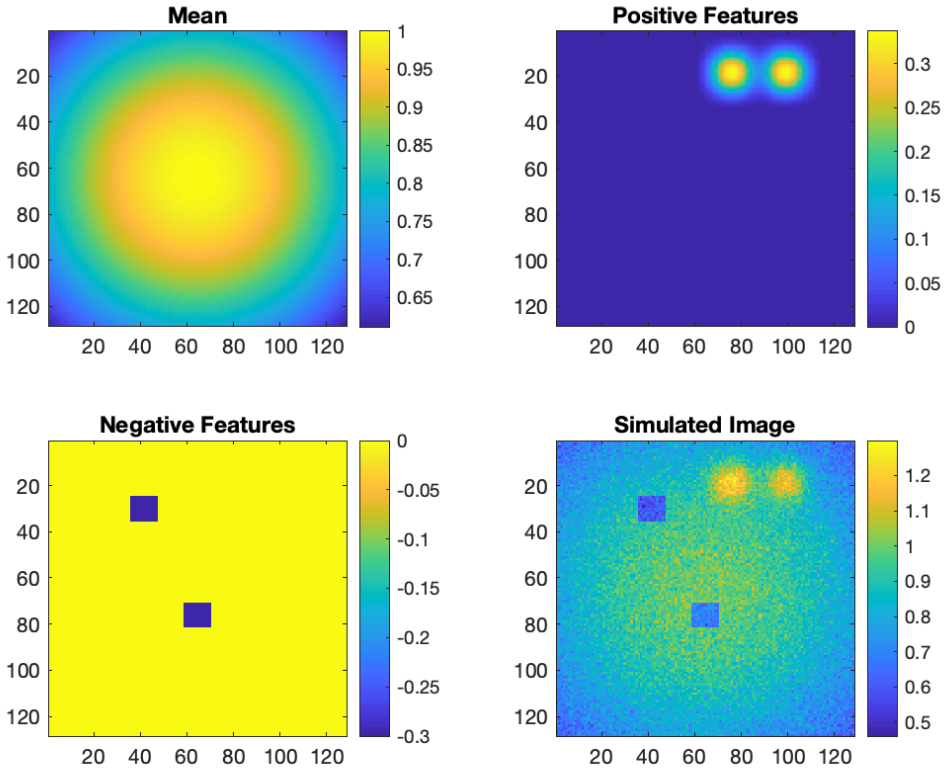

Figure 3.8: Scenario 2 – a sample simulated image with $\delta_p = 0.6$ and $\delta_n = -0.3$

3.4.2 Scenario B - different type of positive and negative features

In this scenario, positive features are generated using $A_p = B_p A_{ps} B_p^T$, where B_p is a cubic B-spline basis with 12 knots and A_{ps} is a sparse matrix of size 13×13 in which two randomly selected entries are replaced with a small number $\delta_p > 0$. The negative features are generated using $A_n = B_n A_{ns} B_n^T$, where B_n is a linear B-spline basis with 12 knots and A_{ns} is a sparse matrix of size 11×11 in which two-randomly selected entries are replaced with a small number $\delta_n < 0$. A sample simulated image along with mean, positive features, and negative features is shown in Figure 3.8.

In the experiments carried out in this scenario, δ_n is fixed at -0.3. We then try different positive feature magnitudes δ_p. The recovery square root mean square error performance

metrics for SSD (PP) and PFD are presented in Figure 3.9. As it is clear from the Figure 3.9, e_M for SSD (PP) is always higher than PFD and experiences a slight jump after $\delta_p = 0.9$, but it remains almost constant for PFD. Both methods have almost same values for e_{A_p}. The performance metric e_{A_n} remains negligible for PFD, but it is always significant for the SSD (PP). This is due to the identifiability issue faced by SSD (PP) as discussed later. Combining the performance in the cases of positive and negative features, the metric clearly shows that PFD performs much better in comparison to SSD (PP). As it can be seen in Figure 3.10, PFD maintains perfect precision and recall scores for both positive and negative features when δ_p is varied. Although SSD (PP) scores a perfect recall when δ_p is varied from 0.3 to 0.9, it has a pretty low precision because it carries a much higher amount of unnecessary features along with the actual negative features. This is illustrated with the help of Figures 3.11 and 3.12. Starting $\delta_p = 1.2$, SSD (PP) is unable to detect negative features. This is illustrated with the help of Figures 3.13 and 3.14. Nick local thresholding recalls positive features but with very poor precision. It is also unable to recall negative features completely. Global thresholding is unable to recall positive and negative features with precision. The sample visual results for Nick local and global thresholding are shown in 3.15. In scenario 2 as well, PFD performs better than the existing methods. Overall, PFD outperforms benchmark methods in the simulation study.

3.5 Case Study

As discussed in Section 3.1, CT image segmentation is an important task in the medical domain as it helps in computer-aided disease diagnosis and surgery planning. To illustrate the functionality of our proposed model and algorithm, we consider a two dimensional cross-sectional CT image of the human heart in the annulus region, as shown in Figure 3.1. This image has special medical importance as it gives information on calcification and aortic root dimensions. We are interested in calcification (high-intensity region) detection and soft tissues (low-intensity segment) extraction. The proposed PFD model is employed

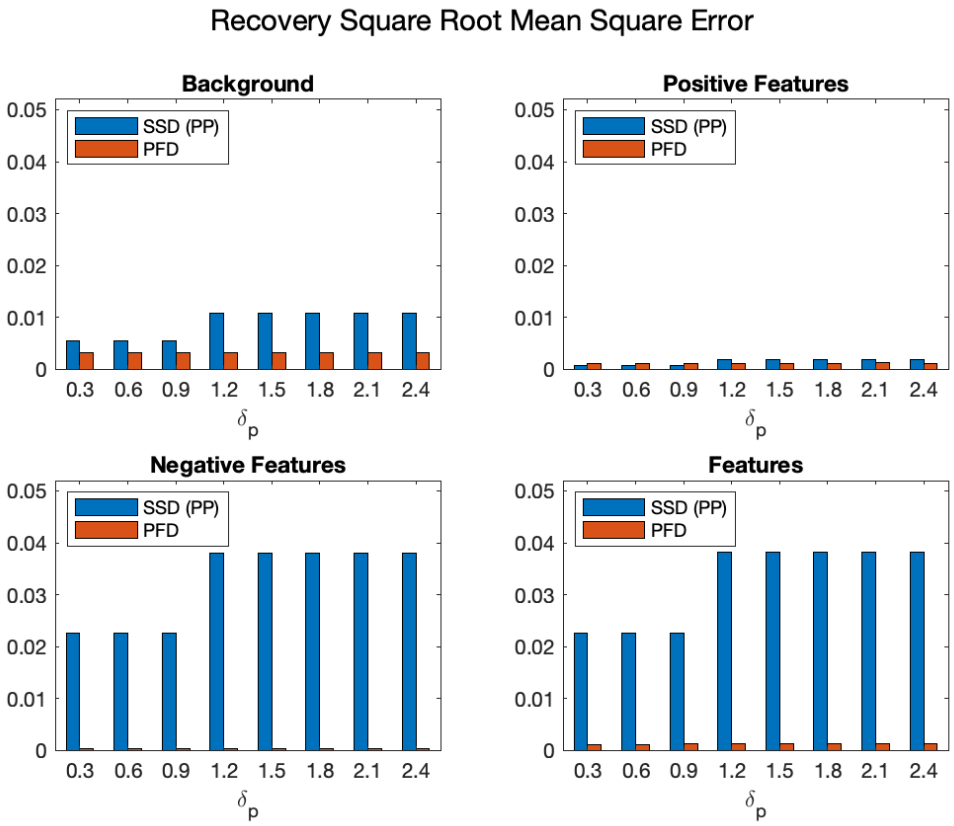

Figure 3.9: Scenario 2 – recovery square root mean square error results showing comparison between SSD (PP) and PFD

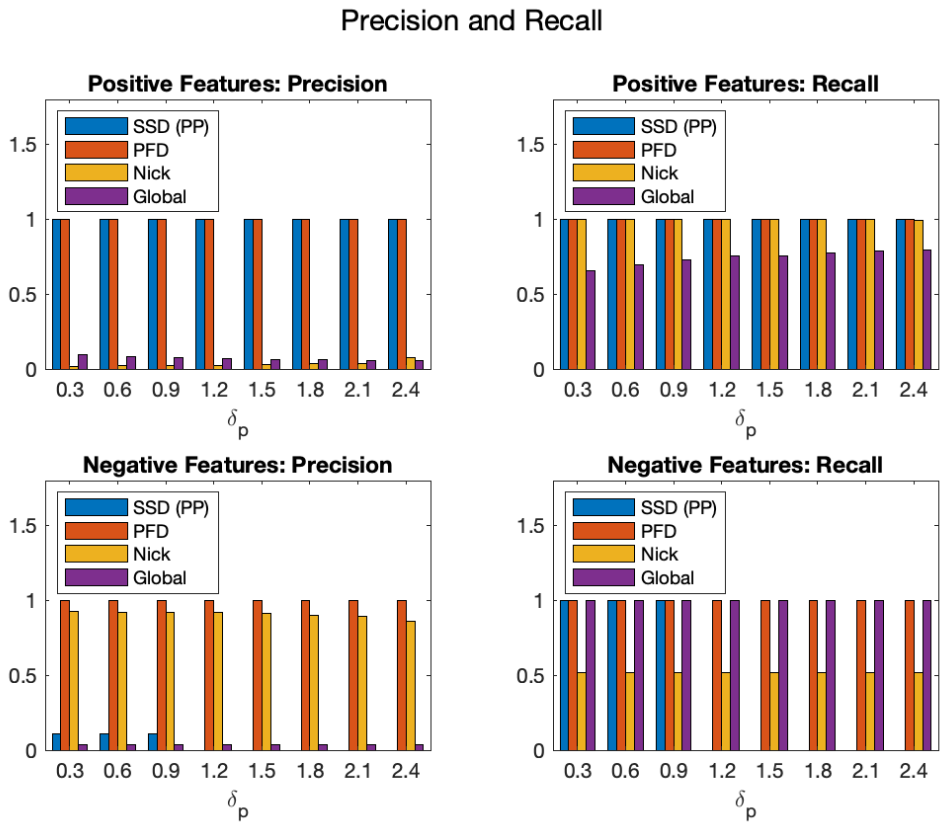

Figure 3.10: Scenario 2 – precision and recall results showing comparison among SSD (PP), PFD, Nick local thresholding, and global thresholding

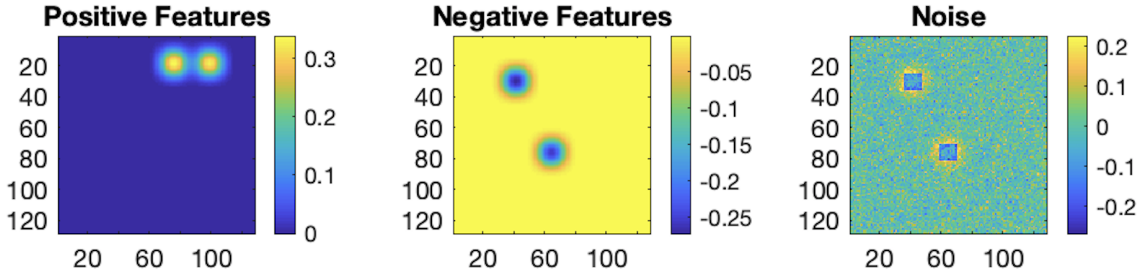

Figure 3.11: Scenario 2 – same basis issue faced by SSD (PP), when $\delta_p = 0.6$

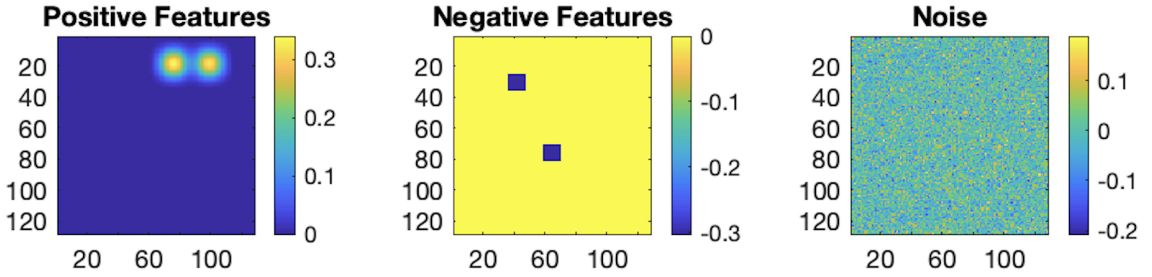

Figure 3.12: Scenario 2 – different basis successfully introduced using PFD, when $\delta_p = 0.6$

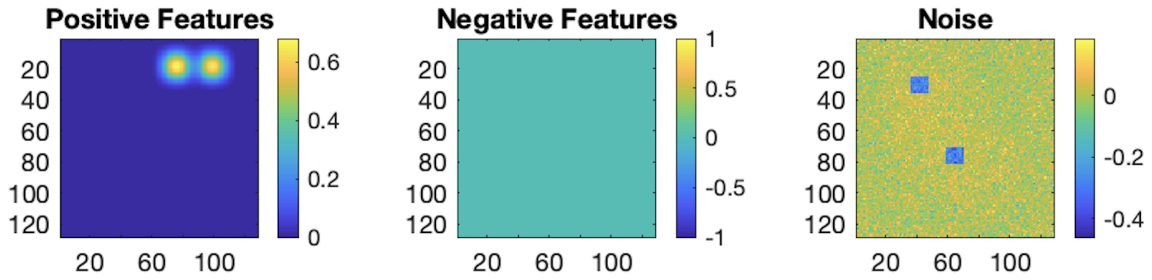

Figure 3.13: Scenario 2 – identifiability issue faced by SSD (PP), when $\delta_p = 1.2$

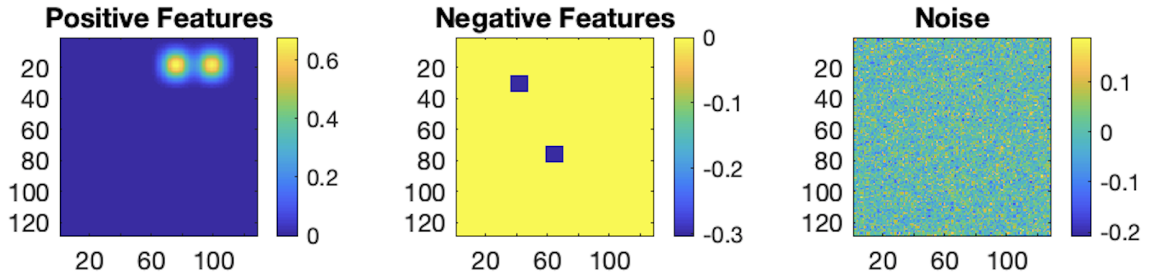

Figure 3.14: Scenario 2 – identifiability issue successfully dealt with PFD, when $\delta_p = 1.2$

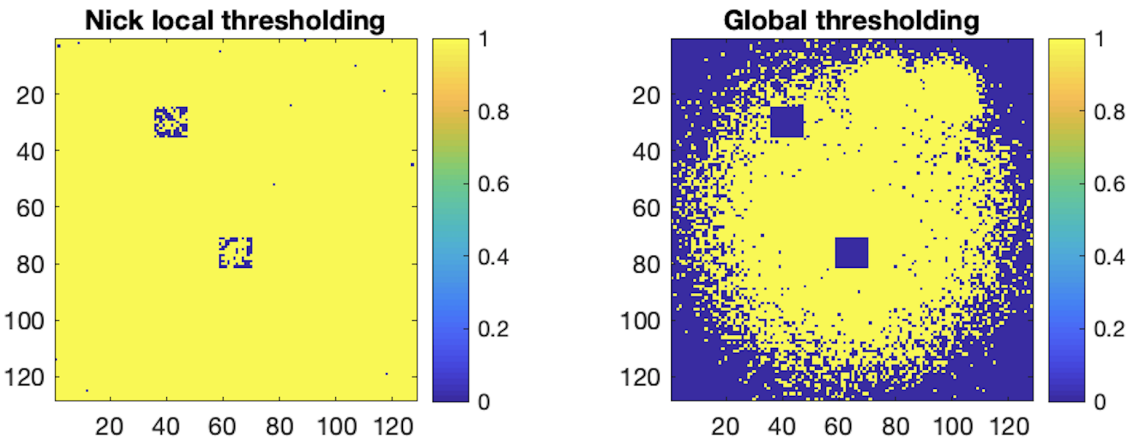

Figure 3.15: Scenario 2 – features detected by Nick local thresholding and global thresholding, when $\delta_p = 0.6$

to accomplish this task.

While applying the proposed PFD approach to the aforementioned image with size (101×101), an identity basis is used for detecting calcification as it is randomly scattered over the mean. Although a cubic B-spline basis could have been used for the extraction of soft tissues, that contain aortic valve, since they form a clustered region, a simpler option, i.e., an identity basis is opted as it is found to be successful. A cubic B-spline with 6×3 knots for the background is selected. Hyper-parameter λ is selected so that it minimizes the GCV as well as ensures convergence of the ADMM algorithm. Other hyper-parameters (ρ, γ_p, γ_n) are dynamically tuned as discussed in Section 3.3.2. To compare the performance, other benchmarks such as SSD with post-processing, Otsu's global thresholding, and Nick local thresholding, are also applied. The results of different methods are shown in Figure 3.16.

Now, we discuss the performance of the proposed approach. As it is clear from the second plot in the top row of Figure 3.16, the proposed approach is able to extract both the calcification and the soft tissues containing aortic valve structure. In the second row of Figure 3.16, the first plot shows post-processed results of the SSD. As far as calcification detection is concerned, there are too many false positives detected by this method. More-

over, the structure of the extracted soft tissues is not as good as the one detected by the PFD. The second plot of the second row and plot in the third row show results of the global thresholding and Nick thresholding, respectively. Since the global thresholding selects a single threshold based on intensity distribution, it clearly fails to extract and separate the calcification and the aortic root. The results of Nick thresholding mostly show the soft tissues region but that too with some false positives. Overall, the performance of the proposed PFD model is better than the benchmarks.

Furthermore, the results obtained using the proposed PFD model have clear medical importance. Our model clearly identifies the amount and location of the calcification. It helps doctors estimate the post-TAVR complication in terms of PVL. From the extracted soft tissues, aortic valve structure can easily be seen. This helps in selecting the appropriate size of the artificial valve to be implanted as a part of the TAVR surgery. This can again help in reducing the chances of post-TAVR complications in terms of PVL and stress induced in the aortic tissues. Furthermore, the proper segmentation of different features in the CT image helps develop a digital model to be used as an input to the 3D-printing which has shown the ability to assist doctors in TAVR surgery planning [27]. Overall, the effectiveness of the proposed approach is quite convincing. Moreover, such an approach can be utilized for image segmentation tasks in other applications such as defect detection in manufacturing.

3.6 Conclusion

Computed tomography image segmentation is an important task in the medical domain. It is increasingly used for disease diagnosis and surgery planning. Aortic stenosis is one such heart disease where CT image is utilized for different purposes such as initial diagnosis, clinical decision making, and surgery planning. Through the heart CT image, doctors explore the cause and extent of disease. For example, they try to locate calcification and its amount. Also, they are interested in understanding the structure of the aortic valve. With

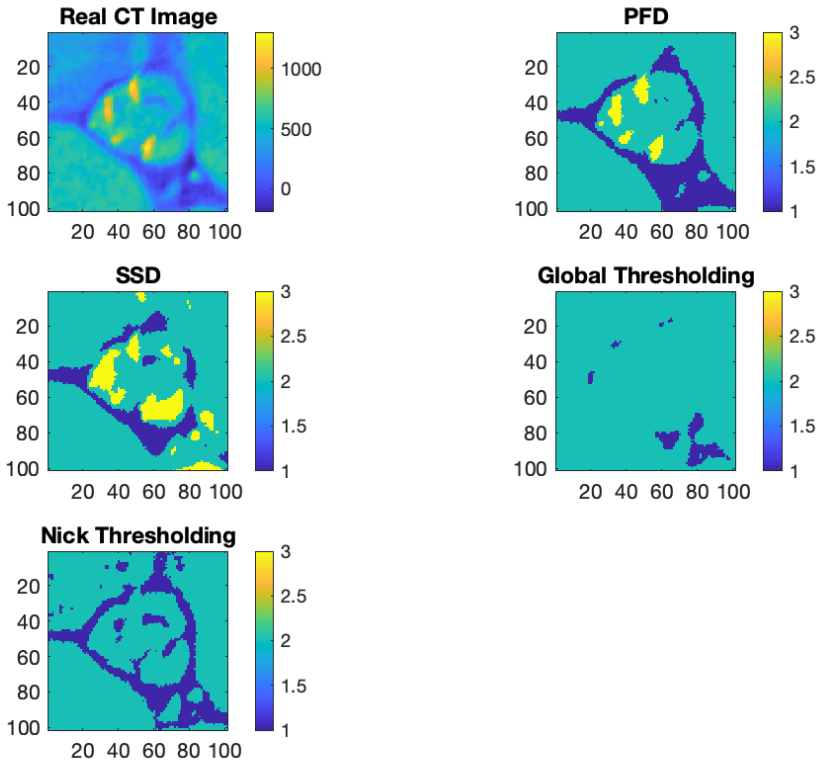

Figure 3.16: Calcification and soft tissues detection using PFD, SSD, global thresholding, and Nick local thresholding

this knowledge, they are able to better understand the patient's condition and accordingly plan the treatment.

The major contribution of this work is to propose an approach to automatically quantify and locate the calcium, and extract the aortic valve, without the need for manual intervention. The proposed approach is flexible enough that it can be used for segmenting important features out of the CT image of other human body parts as well. It models image as if it consists of four components - mean, positive extreme features, negative extreme features, and noise. The mean is further approximated using a set of smooth bases. Extreme features are also modeled using a set of appropriate bases. The approach treats background as smooth and extreme features as sparse. To estimate the parameters of the proposed model, an optimization problem is framed. This optimization problem is solved using a popular algorithm - alternating direction method of multipliers (ADMM). Within ADMM, an efficient proximal gradient method is utilized. Hyper-parameters selection strategy is also discussed in detail. The performance of the proposed approach is tested using an extensive simulation study and real case study. It is found that the PFD model outperforms benchmark methods in both the simulation study and the real case study. In the real case study, the results achieved by the proposed approach are not only precise but also of practical importance. The results are expected to play an important role in clinical decision making and surgery planning. Moreover, results can also act as input for the digital model used in 3D-printing technology for manufacturing virtual aortic valve.

CHAPTER 4

CONVOLUTIONAL NEURAL NETWORK-ASSISTED ADAPTIVE SAMPLING FOR SPARSE FEATURE DETECTION IN VIDEO AND IMAGE DATA

4.1 Introduction

4.1.1 Motivating Example 1

Video data is ubiquitous. From social media to education, video data can be found everywhere. From capturing memories to delivering lectures, videos are used in many ways. One recent use of video data is seen in the form of interview sessions. Video interviews are expected to play an important role in the hiring process [60]. Video-based screening interviews are expected to help in shortlisting candidates for actual job interviews. A video-based resume is expected to become a part of the job application process because it allows candidates to showcase their personality and communication skills. Naturally, in such a scenario, candidates will be required to practice video interviews. Recently, an online interview practice platform was launched. This platform allows a user to conduct a practice interview and this practice session is recorded in the form of a video. The platform provides the user with evaluation based on the interview. Evaluation can be made at different levels such as behavioral and technical.

For the evaluation at a behavioral level, certain facial expressions can be analyzed because they are considered to be most predictive of a candidate's hirability chance. For example, according to psychologists, emotions such as anger, disgust, fear, and sadness could depict a candidate's nervousness and state of stress which in turn could affect the hirability decision. An ideal way to evaluate at the behavioral level would be to watch the whole video and determine in which frames of the video the candidate is looking nervous. This approach would require a lot of manpower. Also, it would be time-consuming

and expensive. Recently, researchers have developed predictive models to predict apparent personality traits and hirability scores based on audio, visual, and language features extracted from video interviews. However, these models require the whole video for the analysis which can be time-consuming. Moreover, these models do not pinpoint video frames where features responsible for lowering a candidate's chance to get hired, can be found. Furthermore, video data is often not completely available for server-side processing due to bandwidth reasons.

4.1.2 Motivating Example 2

In manufacturing systems, product inspection is an important part of quality control. Classifying a product as defective or non-defective is critical from the quality control point of view. For a majority of products, images are taken during the various steps of their manufacturing process as well as at the end of the process. Analyzing such images at different stages of the process can help control the quality of the product.

Quality control using images has become common, but this application brings new challenges. First, large images may be too slow to process in an online setting. This raises a need to efficiently process the images so that defect/anomaly can be detected quickly. Second, due to practical limitations such as cost of sensing, storage, and bandwidth, it is difficult to transfer/receive the whole image in some applications. Only portions of the image can be processed to perform anomaly detection quickly. This type of situation is common in online sensing. Third, images with complex background and features have become common. Often, anomaly and background are indistinguishable (e.g., refer to a sample shown in Figure 4.1) with almost no pixel intensity difference between them. Moreover, in some scenarios, one is interested in a particular kind of anomaly, thus a selective anomaly detection is needed. Existing methods either require the entire image for processing or are unable to deal with the image having complex and indistinguishable anomaly-background structure.

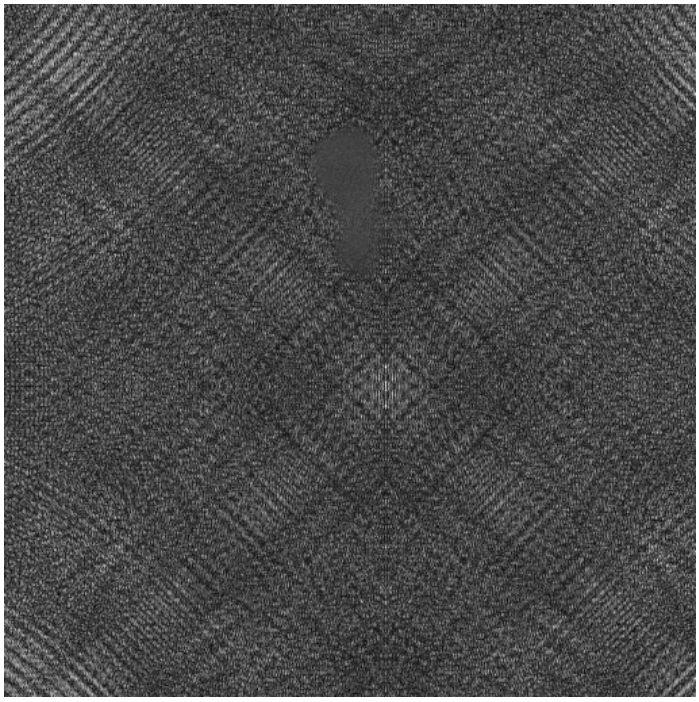

Figure 4.1: Product with a ”diffuse” kind of defect [61]

4.1.3 Problem Definition and Proposed Approach

Based on the motivating examples and assuming features are sparsely present, the problem is how to smartly sub-sample a video for quick behavioral trait/emotion identification and an image for quick anomaly detection. Instead of processing all the frames of the video or the patches in the image, random sampling can be performed and frames/patches can be selectively processed so that features can be detected quickly. Advanced sampling techniques such as sequential maximin sampling can also be utilized. However, random sampling doesn't sample uniformly and sequential maximin sampling can sample uniformly but cannot concentrate around the frames/patches we are interested in. As a result, these sampling techniques are not expected to work efficiently. Recently, an adaptive sampling has been proposed which can sequentially sample frames/patches and concentrate at regions of interest for more information on features [62].

During sequential sampling, a model is needed to distinguish between desired features and background. Recent developments in deep learning have made image and video analy-

sis much more feasible. The convolutional neural network (CNN) model can be developed using existing images. For example, a) a CNN model can be trained using anomalous and non-anomalous patches in the case of defect detection in the manufacturing domain, b) a CNN model can be developed using training images containing different human emotions such as angry, happy, and neutral. Such models can be used to focus on regions of interest in images and videos during sampling.

In this work, we propose a novel sampling approach for sparse feature detection in image and video data. The proposed sampling approach is based on a combination of an adaptive sampling method, which allows only a few patches/frames to be observed and a convolutional neural network, which helps distinguish between background and features conveniently. While adaptive sampling helps explore and exploit the different patches of the image or frames of the video in a smart manner to discover features, the convolutional neural network directs the sampling to focus on the features of interest. The proposed approach is shown to have a large potential for the application where complete image sensing and video processing is constrained due to practical reasons and quicker feature detection is required.

The rest of the paper is organized as follows. Motivation and problem definition have been put forth in this section. Relevant literature is presented in Section 2. The proposed approach comprising of adaptive sampling and CNN is detailed in Section 3. The performance of the proposed approach is evaluated using case studies in Section 4. Finally, conclusions are made in Section 5.

4.2 Related Work

4.2.1 Analysis of Interview Videos

Research efforts on analyzing interview videos for candidate evaluation are limited. In [60], researchers have analyzed online conversational audiovisual resumes to form job-related first impressions. Audio and visual modalities are used to extract features. Correlation

analysis is performed to check the relationships between nonverbal behavior and hirability. In [63], researchers have investigated the relationship between stress impressions and the interview outcome (being hired or not), the audio and the visual cues, and the electrodermal activity measured through wearable technologies. They found stress impressions to be negatively correlated with hirability ratings. Also, their results show that stress impressions are better predicted by visual features than audio ones. In [64], researchers have proposed a prediction model relating features such as prosodic, lexical, and facial, with interview ratings and other interview-specific traits such as excitement, friendliness, engagement, and awkwardness. They also propose a way to make recommendations for the person being interviewed so that (s)he can improve his/her hirability score. In [65], authors utilize audio, face, and scene features computed from video interviews as inputs to predict apparent personality traits and a categorical response encoding whether the subject will be invited to the interview. Features are fed into feature-specific regressors to predict responses. Base learners are then stacked together to form an ensemble of decision trees, which produces quantitative outputs. A single decision tree combined with a rule-based algorithm then produces interview decision explanations based on quantitative results.

Recent years have seen deep learning-based models achieve great success for various applications. Such models have also been proposed for first impressions analysis from video interviews/resumes/blogs. The advantage of such models is that they do not require an explicit feature extraction procedure to be performed on the input data fed to these models. In [66], researchers have developed a deep residual network model establishing a relationship between audiovisual streams and personality traits. Also, they explore language data to predict personality traits. Additionally, they utilize audiovisual and/or language data to predict interview recommendations. In [67], authors identify audiovisual information exploited by their deep residual network to recognize personality traits. This information is useful for explaining recommendations made by the model.

The major focus of the aforementioned works is predicting the first impressions, per-

sonality traits, and hirability. Some of these work has also focused on the explainability of the prediction models. However, these approaches require processing the whole video, which can be time-consuming, for tasks such as feature extraction and modeling. In the test setting, such methods require the availability of complete video for the server-side analysis. This may not be possible due to bandwidth issues. Furthermore, these methods do not focus on locating the video frames where features responsible for lowering the chance of hirability can be found. Detecting features and their location is important because interviewers can then closely focus on the located frames. Also, this could help practice platforms to recommend candidates where to improve.

4.2.2 Anomaly Detection in Textural Images

Anomaly/defect detection in images is a challenging visual recognition problem arising in several stages of the manufacturing process. Toward this end, researchers have carried out a considerable amount of research to develop approaches for anomaly detection in images. We mainly focus on reviewing the parts of the research which is related to defect detection in textured surfaces. [68] has grouped textural surface defect detection methods into four main categories – (a) statistical, (b) structural, (c) filter-based, and (d) model-based. We mainly focus on reviewing model-based methods. In [69], the researcher has developed an approach for the segmentation of local textile defects. Their approach consists of feature extraction for every pixel by exploiting the gray-level arrangement of its neighboring pixels, dimensionality reduction using principal component analysis, feed-forward neural network model to determine the defect, and post-processing results using median filtering. In [70], researchers have developed an approach to localize defects in random color textures. A sample of defect-free images is used and patches are generated using those images. A mixture model is applied to these patches to obtain texture exemplars. The defect in the inspection image is detected by measuring the likelihood of each patch. A low likelihood implies a possibly defective region. In [71], researchers have proposed a

framework comprising of feature extraction using Structure Multivector and Harris feature detector, and classification model based on evolutionary reinforcement learning to detect defects in visual inspection images. In [72], researchers have developed an offline training and online evaluation framework to detect surface defects on textured surfaces. The offline training module considers a weakly labeled data set consisting of non-defective and defective surface images. They randomly select patches from each image and give them the corresponding labels. Based on their grayscale values, patches are further converted into statistical features. A neural network model is trained based on the data set comprising of feature vectors and their corresponding labels. In the online evaluation, they propose to analyze the whole image in a sliding window (with a stride) fashion and in each window, statistical features are computed and then the model predicts the presence or absence of a defect.

The limitations of these papers are as follows. First, some methods require a separate feature extraction step at the beginning. Second, in some approaches, post-processing of results is required. Third, in some cases, in the testing phase, one needs to traverse the entire image patch by patch to detect the sparse anomaly, which may be too time-consuming in the case where the image is being sensed in real time through a monitoring device.

Last decade has witnessed an increased use of deep learning due to advancement in computational ability. CNN has gained popularity in computer vision. It has the capability to perform feature extraction and learn a classification model in a single framework. It has shown the capability to overcome the first limitation mentioned in the previous paragraph. In [73], researchers have developed a unified CNN-based framework for segmentation and detection of surface anomalies. The developed network has the capability to learn anomaly representations using a small set of coarsely labeled training examples. In [74], researchers presented a deep learning-based approach for the detection and segmentation of surface cracks. The network architecture design enables model training using a small number of samples. In [75], researchers proposed an efficient deep learning-based method for pixel-

wise surface defect detection and segmentation. Their approach consists of three stages – (a) segmentation stage, (b) detection stage, and (c) matting stage. The major limitation of these methods is that in the testing phase, the entire image is required for the analysis.

4.2.3 Sampling Techniques

A significant amount of research exists in the area of the sequential design of experiments (SDOE). Sequential sampling strategies have been developed for spatial profile data. There are two categories of such techniques – model-based and distance-based [76]. However, the major limitation of SDOE techniques is that their focus is mainly on improving the overall model fitting of the spatial profile [77, 78]. Recently, researchers have developed a sampling strategy named Adaptive Kernelized Maximum Minimum-Distance (AKM^2D) combining maximin design of experiment approach for the exploration of the space and the Hilbert Kernel approach for the targeted sampling in desired regions of the space [62]. In this work, we combine the CNN model with (AKM^2D) to detect the emotions in the videos and the sparse anomalies in the images.

4.3 CNN-assisted Adaptive Sampling-based Feature Detection

In this section, we discuss the proposed approach to detect the features in videos and images. The proposed approach comprises of two parts: the adaptive sampling technique and the CNN model. The approach is summarized in Figure 4.2. First, initial patches/frames are sampled using a space-filling design (e.g. maximin distance [79]) to explore the entire sampling space. Then, the CNN model is used to assign a probability of being a desirable/non-desirable feature to each one of those patches/frames. Desirable patches/frames get a high probability than undesirable ones. Then, based on these initial points, subsequent points are chosen by using the adaptive sampling criterion which balances between the exploration of the entire space and the exploitation of the space near desired regions. After the sampling criterion chooses the next point, the CNN model is used to find its probability of

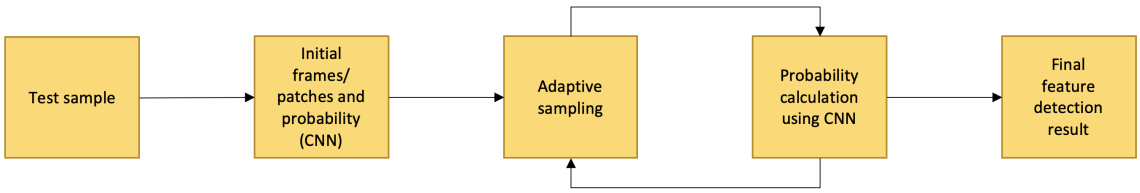

Figure 4.2: Approach overview

being a desired feature. Finally, this procedure is repeated until the constraint in terms of the maximum number of sampling points is met.

4.3.1 Adaptive Sampling

Adaptive sampling takes place in a single dimension in the case of a video, whereas it is being performed in 2-dimensions in the case of an image. As mentioned earlier, in the case of a video, sampling is performed at a frame level, whereas in the case of an image, it is performed at a patch level. We first explain the adaptive sampling for the simpler case, i.e., 1-D and then we discuss it for the 2-D.

Assume the length of the video is m defined in terms of total number of frames. One-dimensional sampling space $[0, 1]$ is used. The frames are constrained to be on a 1-D fine grid defined as $\mathcal{G}_{v,m} = \{\frac{i}{m-1} | i = 0, 1, ..., m-1\}$. Let n be the number of frames sampled so far. Further, suppose those frames are located at $\mathcal{A}_{v,n} = \{f_k \in \mathcal{G}_{v,m} | k = 1, ..., n\}$. Suppose $p_a(f_k)$ denote the estimated probability of the frame containing the desired feature. The following criterion developed by [62] is used to find the next sampling frame:

$$f_{n+1} = \underset{f \in \mathcal{G}_{v,m}}{\operatorname{argmax}} \ \psi_n(f)(\phi_n(f))^{\lambda}, \tag{4.1}$$

where $\psi_n(f)$ is the estimated distribution of desired features and $\phi_n(f)$ is the regularization term. Maximizing $\psi_n(f)$ enables exploitation which means that the next proposed sampling frame is selected near one of the desired feature regions, whereas $\phi_n(f)$ enables exploration and prevents sampling frames getting too close to each other. In the

above criterion, $\psi_n(f)$ is defined as a mixture distribution consisting of Gaussian distributions and a uniform distribution. Each Gaussian distribution is centered at an observed desired feature. Uniform distribution accounts for unobserved desired features in the entire sampling space. Specifically, $\psi_n(f) = (\sum_{k=1}^{n} p_a(f_k)K_\sigma(f, f_k) + u)$ where $K_\sigma(f, f_k) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{||f-f_k||^2}{2\sigma^2})$ is the 1D-Gaussian kernel centered at a desired feature frame f_k, which represents the clustered structure of the features; $p_a(f_k)$ plays the role of mixture weight for the Gaussian distribution; u follows uniform distribution. The regularization term $\phi_n(f)$ is defined as $\phi_n(f) = \min_{f_k \in \mathcal{A}_{v,n}} ||f - f_k||$, where $||\cdot||$ is the l_2 norm, to introduce space-filling property in the sampling criterion. By substituting $\psi_n(f)$ and $\phi_n(f)$, the sampling criterion given in (4.1) can be rewritten as follows:

$$f_{n+1} = \underset{f \in \mathcal{G}_{v,m}}{\text{argmax}} \left\{ \left(\sum_{k=1}^{n} p_a(f_k)K_\sigma(f, f_k) + u \right) \left(\min_{f_k \in \mathcal{A}_{v,n}} ||f - f_k|| \right)^{\lambda} \right\}. \tag{4.2}$$

Since the feature detection is required to be fast, the adaptive sampling criterion needs to propose the next sampling frame quickly. This requires an efficient way to solve the optimization problem in (4.2). The first item in the exploitation term of the adaptive sampling criterion can be efficiently computed using the product of a matrix and a vector. That is, the exploitation term can be computed as $\Psi_n = K_\sigma^T P_A + u1_{m\times 1}$, where $K_{\sigma,ij} = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{||i-j||^2}{2\sigma^2(m-1)^2})$, and $P_{A,i1} = p_a(\frac{i}{m-1})1(\frac{i}{m-1} \in \mathcal{A}_{v,n})$ are the ij^{th} element of the matrix K_σ and i^{th} element of the vector P_A, respectively. Here, $i, j = 0, 1, ..., m-1$. The indicator function $1(x)$ is defined as $1(x) = \begin{cases} 1 & \text{if } x \text{ is true} \\ 0 & \text{if } x \text{ is false} \end{cases}$, and $1_{m\times 1}$ is an m by 1 vector of 1s. It can easily be shown that $\phi_n(f)$, $f \in \mathcal{G}_{v,m}$ can be recursively updated by $\phi_n(f) = \min(\phi_{n-1}(f), ||f - f_n||)$, $f \in \mathcal{G}_{v,m}$. Therefore, the optimization problem in (4.2) can be efficiently solved using the Algorithm 2.

Algorithm 2 One-dimensional Adaptive Sampling

Initialization:

Initial n_{init} sampling based on max-min distance design

for $n = n_{init}, ..., n_{max}$ **do**

Update $\psi_n(f)$ based on $\Psi_n = K_\sigma^T P_A + u1_{m\times 1}$

Update $\phi_n(f) = \min(\phi_{n-1}(f), ||f - f_n||)$ for $f \in \mathcal{G}_{v,m}$

$f_{n+1} = \text{argmax}_{f\in\mathcal{G}_{v,m}} \psi_n(f)(\phi_n(f))^\lambda$

end

Now, the adaptive sampling for the case of image is discussed. Assume the size of the image is $m \times m$. The image pixels are constrained to be on a 2-D fine grid defined as $\mathcal{G}_m = \{(\frac{i}{m-1}, \frac{j}{m-1})|i, j = 0, 1, ..., m-1\}$. Two-dimensional sampling space $[0, 1]^2$ is used in this case. Assuming the size of the patch selected around a sampling point to be $r \times r$, sampling points are restricted to be on a 2D fine grid defined as $\mathcal{G}'_m = \{(\frac{i}{m-1}, \frac{j}{m-1})|i, j = \frac{r-1}{m-1}, ..., \frac{m-1-r}{m-1}\}$. Let n be the number of points sampled so far. Further, suppose those points are located at $\mathcal{A}_n = \{s_k = (x_k, y_k) \in \mathcal{G'}_m|k = 1, ..., n\}$. Suppose $p_a(s_k)$ denote the estimated probability of the patch being containing the desired feature, which is assigned to the center of the sampling patch. The criterion used to find the next sampling point, similar to the video case, is as follows: $s_{n+1} = \text{argmax}_{s\in\mathcal{G}'_m} \psi_n(s)(\phi_n(s))^\lambda$. The exploitation term in the case of image is defined as $\psi_n(s) = (\sum_{k=1}^n p_a(s_k)K_\sigma(s, s_k) + u)$, where $K_\sigma(s, s_k) = \frac{1}{(\sqrt{2\pi}h)^2} \exp(-\frac{||s-s_k||^2}{2h^2})$ is the 2D-Gaussian kernel centered at a center point of the desired patch, s_k, and u is defined as before. To calculate the exploitation term efficiently, it is re-written as $\Psi_n = K_x^T P_A K_y + u1_{m\times m}$, where $K_{x,ij} = K_{y,ij} = \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{||i-j||^2}{2\sigma^2(m-1)^2})$, and $P_{A,ij} = p_a(\frac{i}{m-1}, \frac{j}{m-1})1((\frac{i}{m-1}, \frac{j}{m-1}) \in \mathcal{A}_n)$ are the ij^{th} elements of the matrices K_x, K_y, and P_A, respectively. Following the case of video, the exploration term in the case of image can be recursively updated as follows: $\phi_n(s) = \min(\phi_{n-1}(s), ||s - s_n||)$, $s \in \mathcal{G}_m$.

4.3.2 Convolutional Neural Network Models

For the case of emotion detection in the interview video example, one of the CNN models proposed in [80] is utilized to detect the emotion of interest in a sampled frame. First, in the sampled frame, the face is detected using Haar feature-based cascade classifiers available in OpenCV, Python [81]. After face detection, a CNN model is used to detect the emotion. The architecture of the CNN model is shown in Figure 4.3. This is a fully-convolutional neural network composed of 10 convolutional layers, ReLUs, batch normalizations, global average pooling, and softmax. This architecture gets rid of the fully connected layer by having the same number of filters in the last convolutional layer as the number of classes followed by global average pooling and softmax to get the prediction probabilities. The model is trained with the ADAM optimizer. Cross-entropy is used as the loss function. The model is trained and tested using the FER-2013 dataset. This dataset contains 35,887 grayscale images where each image belongs to one of the following classes – angry, disgust, fear, happy, sad, surprise, neutral.

For the case of defect detection in manufacturing images, the CNN model is utilized to distinguish between background and anomaly. The CNN model is preferred because of its ability to classify indistinguishable background and anomaly. The model is trained using a set of non-defective patches obtained from a couple of non-defective images and a set of defective patches obtained from the anomalous region of a defective image. To be specific, 1286 patches are randomly sampled from the non-defective image shown in Figure 4.4 and 500 patches are randomly sampled from the non-defective image shown in Figure 4.5. Also, 1286 patches are obtained from the anomalous region of the defective image shown in Figure 4.6. Size of each patch is 16×16. Each patch is given a label – '0' for defective and '1' for non-defective. This set of patches is split into training (75%) and validation (25%) sets.

The architecture of the CNN model is designed from scratch as depicted in Figure 4.7. The model includes one convolutional layer for extracting features and one fully connected

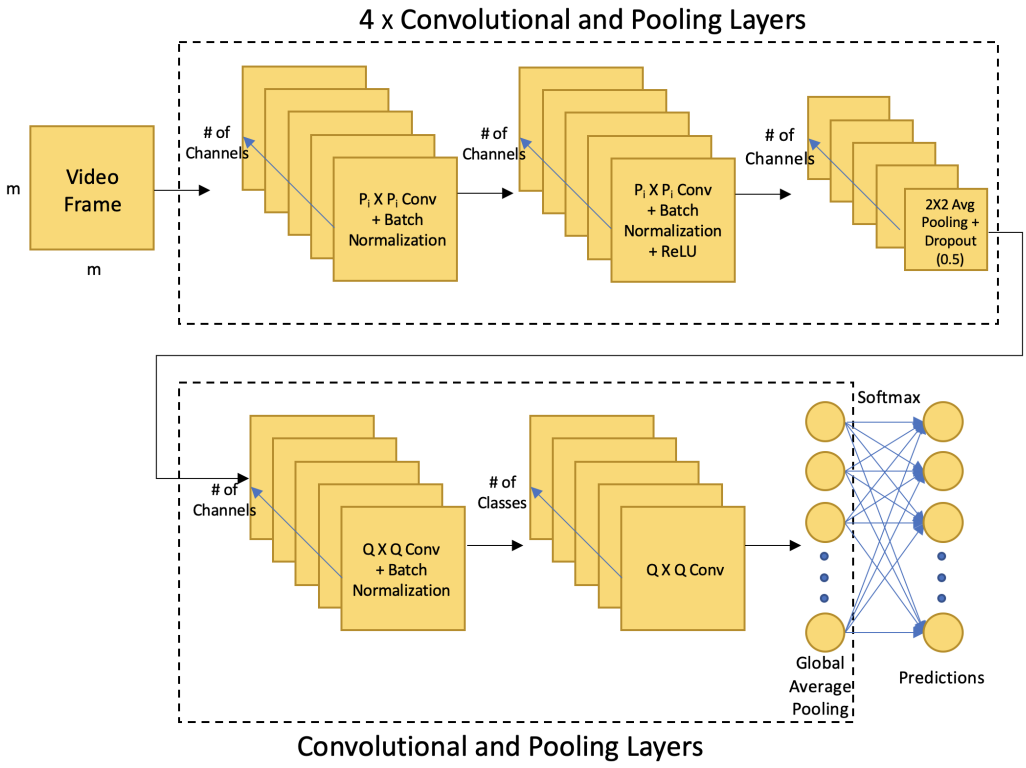

Figure 4.3: CNN model architecture for emotion prediction [80]

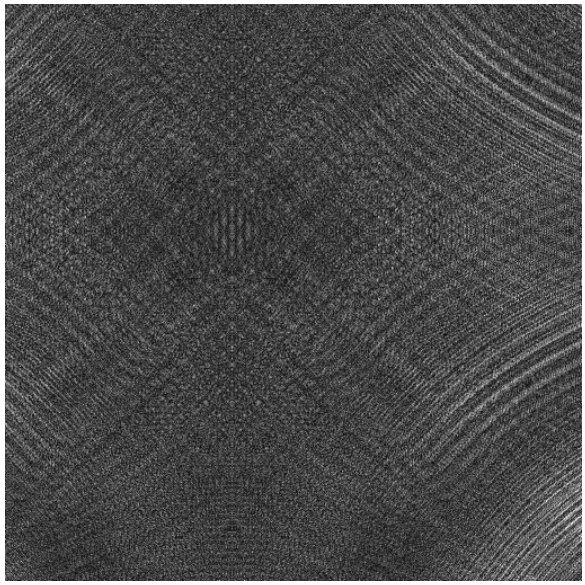

Figure 4.4: Non-defective image sample 1

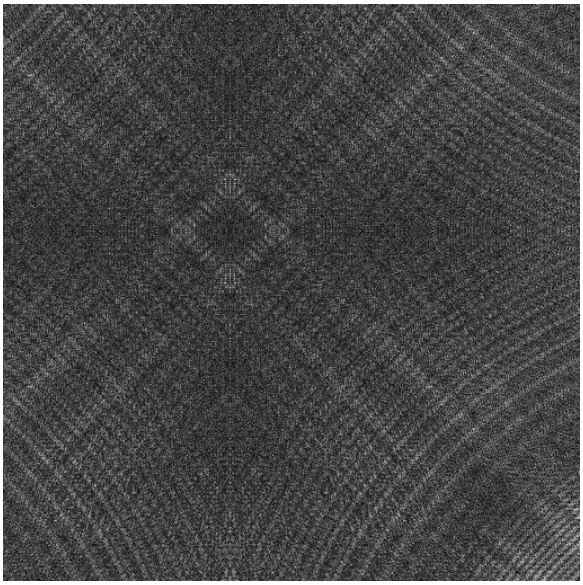

Figure 4.5: Non-defective image sample 2

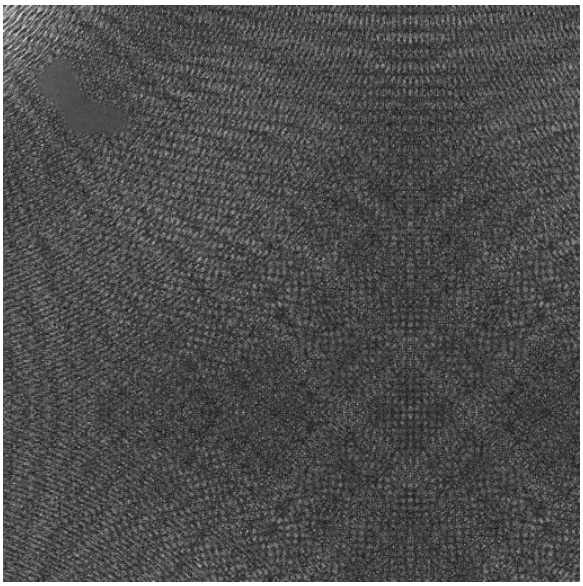

Figure 4.6: Defective image sample 1

layer. The main purpose of the convolutional layer is to extract important features by filtering, so as to enhance the predictive power of the classification model. Different filters could produce different feature maps. The filter size also varies from application to application. In this case, at the convolutional layer, a couple of filters is used. Size of each filter is empirically set to 5×5 with stride 2×2. Since two filters are used, the number of output features from this layer is going to be 2. The size of each output feature is 6×6. Non-linear ReLU operation is performed on the output of the convolutional layer. Next, max-pooling is performed with a pooling window 3×3 and stride 2×2. After performing max-pooling, size of the output becomes $2 \times 2 \times 2$. Then, batch normalization is applied to this output. Finally, the output obtained after convolutional and max-pooling layers is flattened to a vector of size 8×1. This flattened output is fed to a fully-connected layer consisting of 16 hidden neurons and 2 output neurons. A weighted combination of elements of the flattened vector is received at each of those hidden neurons where ReLU is performed. To reduce over-fitting, the dropout technique is applied to the connections between hidden neurons and final output neurons. A weighted combination of output from the hidden neurons is received at the output neurons where softmax function is applied to get the probabilities corresponding to the two classes. The loss function used is cross-entropy and the optimizer used is rmsprop. To judge the performance of the model, accuracy is used as the metric. The CNN model architecture is built using Keras in Python.

The testing accuracy of this model is found to be 98.6%. Once the model has been developed, it can be utilized to distinguish between desirable and undesirable features while adaptive sampling. In this way, the feature detection procedure remains focused.

4.4 Case Studies

In this section, we present case studies using artificial and real data sets to showcase the applicability of the proposed approach.

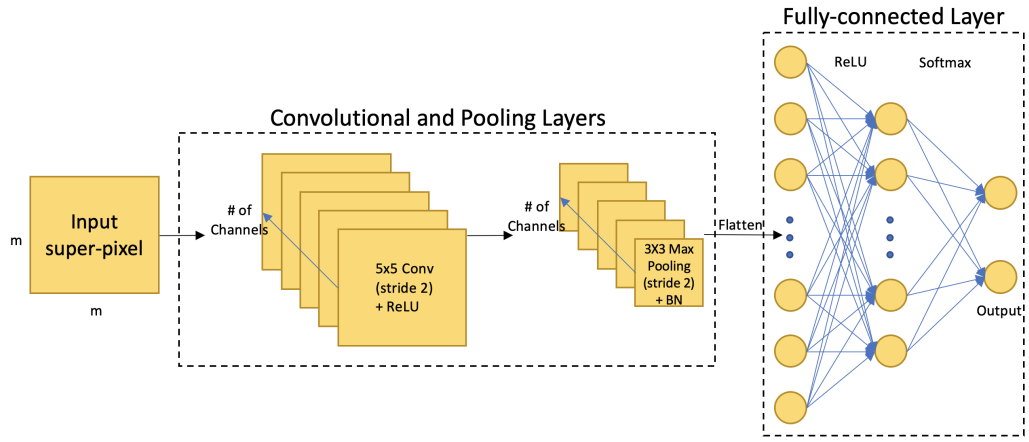

Figure 4.7: Proposed CNN architecture to classify image background and anomaly

4.4.1 Emotion Detection in Interview Video

To demonstrate the proposed approach in the case of a video, we utilize the real interview video obtained from an interview practice platform, a video that is 13.73 minutes long. While evaluating at the behavioral level, certain negative emotions such as anger, disgust, fear, and sadness are indicative of the candidate's nervousness and state of stress according to psychologists. Also, such emotions appear throughout the interview video sparsely. Identifying and quantifying these emotions is important because the presence of such emotions can affect the hirability decision. Identifying the frames with negative emotions is also important. Indeed, if interview practice platforms can transfer this knowledge to the candidate, the candidate could improve on these weaknesses, whereas for recruiters, this knowledge would help reduce their evaluation effort as they can focus on these critical frames. We apply the proposed sampling approach to smartly detect such emotions in the real interview video. To compare the performance of the proposed sampling approach, we utilize two other approaches – sequential maximin and random sampling. The sampling is continued until 50% of the total number of frames have been covered.

The likelihood of finding the negative emotion in a particular frame for the entire video is shown in Figure 4.8(a). It takes 5.26 minutes to get this entire likelihood signal. The negative emotion detected using the proposed sampling approach is shown in Figure 4.8(b).

It is found that the proposed sampling approach is able to recover 98% of the negative emotion and it takes just 2.86 minutes. The frame sampling pattern is shown in 4.8(c). As can be seen in this Figure, the proposed sampling approach concentrates on the zone with a higher likelihood of the desired feature. The negative emotion detected using the sequential maximin sampling is shown in Figure 4.9(b). While taking a slightly lesser time, i.e., 2.67 minutes, than the proposed approach, this sampling approach recovers just 40% of the desired features. Also, its sampling pattern, as shown in Figure 4.9(c) is fairly uniform and doesn't exhibit the exploitation capability. The negative emotion detected using random sampling and the observed sampling pattern are shown in Figures 4.10(b) and 4.10(c). While taking a lower time, i.e., 2.65 than the other two approaches, the random sampling recovers just 46% of the desired features. The fraction of detected negative emotion as a function of sampling iterations is shown in Figure 4.11. It can be seen from this Figure that the proposed sampling approach recovers the desired features quicker than the other two approaches. Clearly, the proposed sampling approach carries out effective sampling to detect the desired features, and performs better than the benchmarks.

4.4.2 Anomaly Detection in Image

To demonstrate the proposed approach in the case of an image, we utilize the sample shown in Figure 4.1, of size 512×512 pixels. As mentioned earlier, this image has an anomaly, which is not easily distinguishable from the background. Detecting such an anomaly is important from a product inspection point of view especially when complete image sensing is restricted due to practical reasons. We apply the proposed sampling approach to smartly detect the anomaly in the image. Two other approaches, namely maximin and random sampling, are used as benchmarks to compare the performance of the proposed approach. The sampling is carried out until 200 sampling points are sampled.

Figure 4.12(a) shows the sampling points pattern obtained as a result of applying the proposed sampling approach to the test image sample and Figure 4.12(b) shows sampling

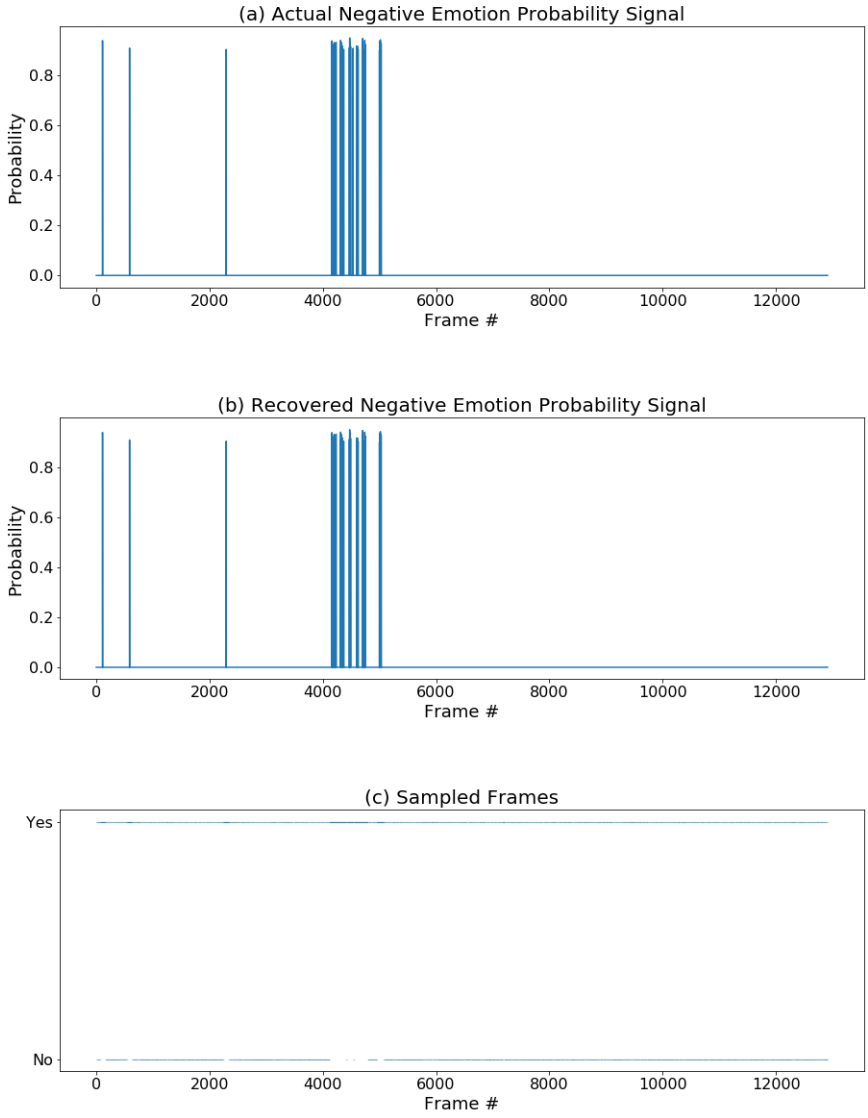

Figure 4.8: Results on emotion detection using adaptive sampling

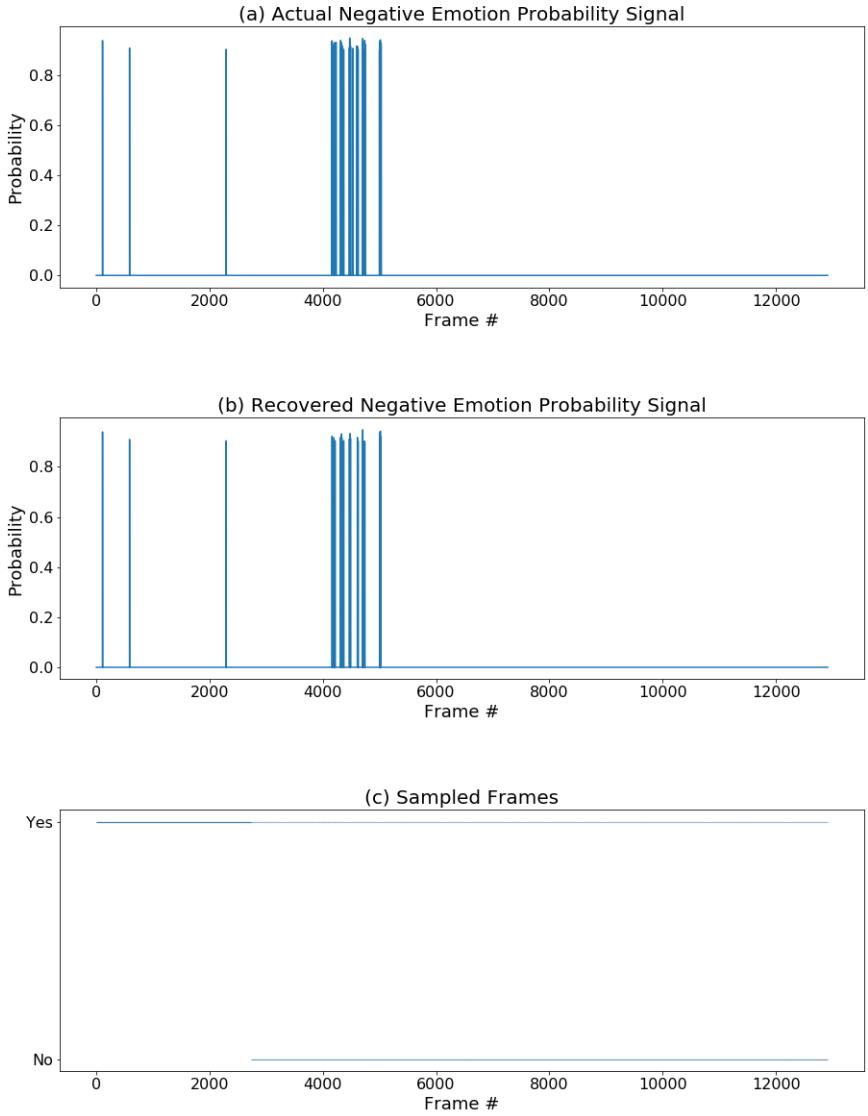

Figure 4.9: Results on emotion detection using maximin sampling

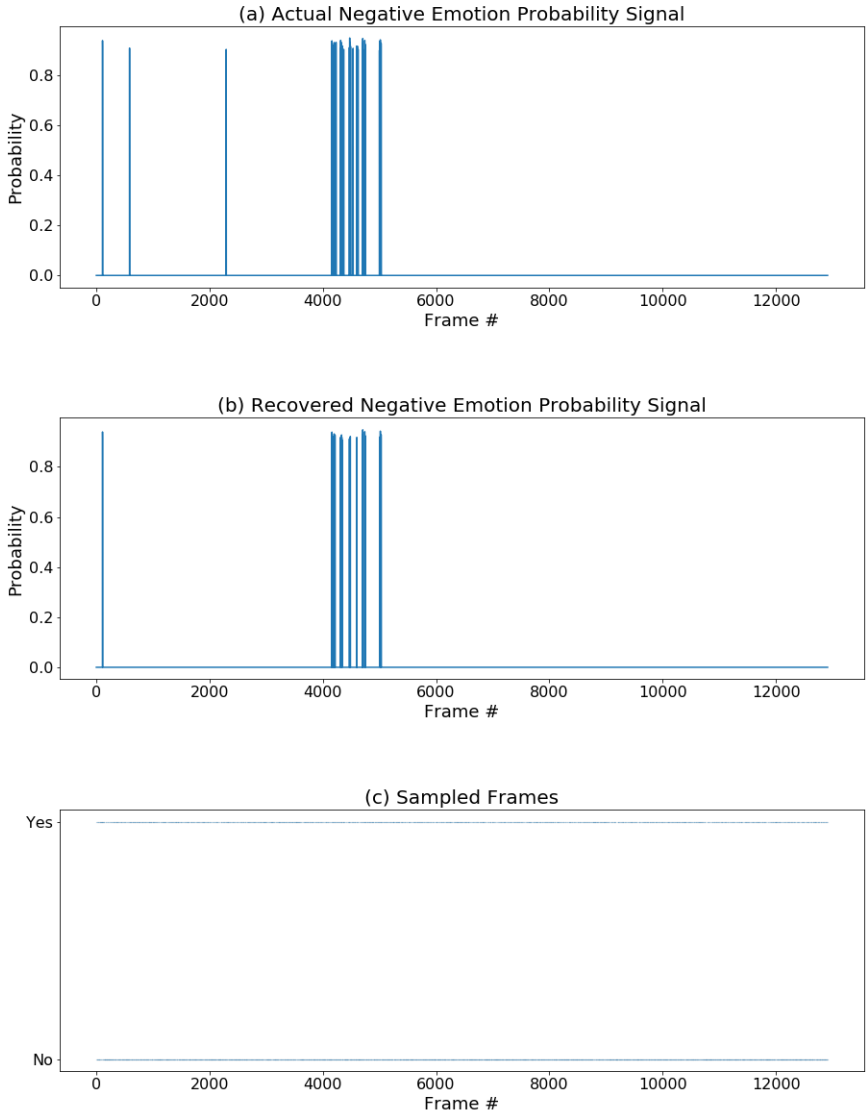

Figure 4.10: Results on emotion detection using random sampling

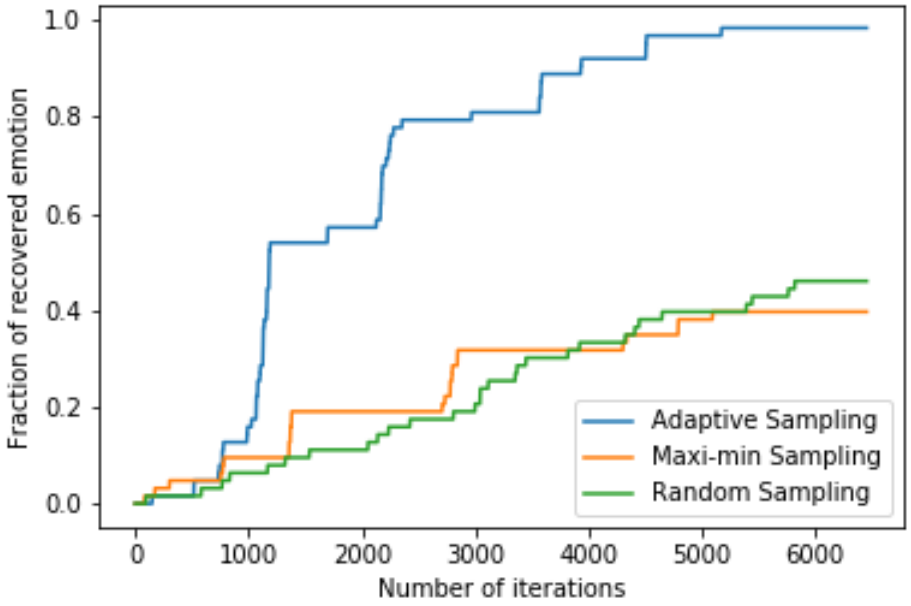

Figure 4.11: Fraction of negative emotion detected by different methods as a function of sampling iterations

patches pattern obtained for the same sample. As it can be from these Figures, the proposed approach exploits the areas where the likelihood of finding the anomaly is higher, by focused sampling. The proposed approach recovers 99.5% of the anomaly with the processing of just 14.3% of the image. When the maximin sampling approach is applied to this case, the sampling points and patches patterns look quite uniform with no explicit focus near the anomalous region, as shown in Figures 4.13(a) and (b), respectively. This approach detects 17.1% of the anomaly by processing 19.5% of the image. Applying the random sampling approach to the image sample yields sampling results as depicted in Figure 4.14(a) and (b). This approach detects 26.1% of the anomaly by processing 17.1% of the image. The fraction of detected desired features (anomalies) as a function of sampling iterations is shown in Figure 4.15. Clearly, the proposed approach performs efficient and effective sampling in comparison to the benchmarks.

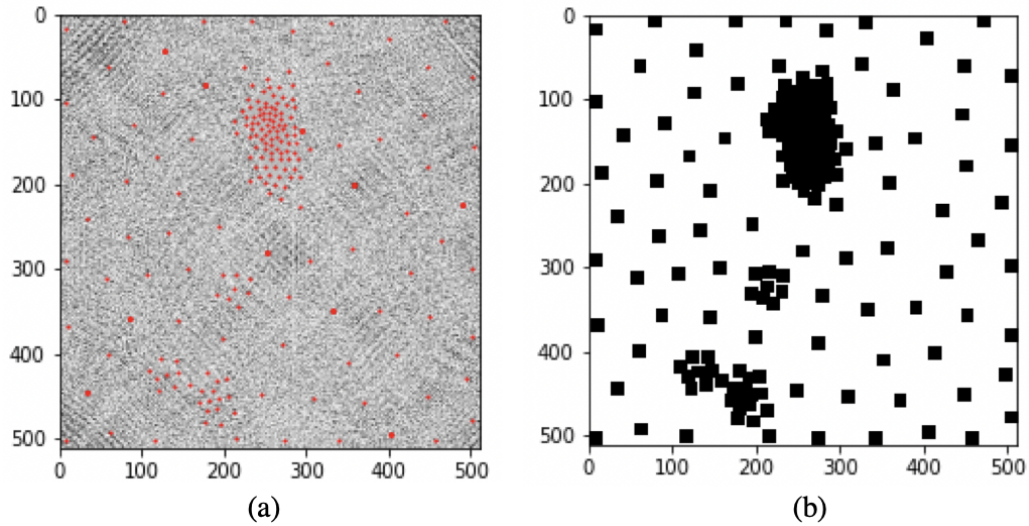

Figure 4.12: Sampling points and patches pattern obtained after applying adaptive sampling

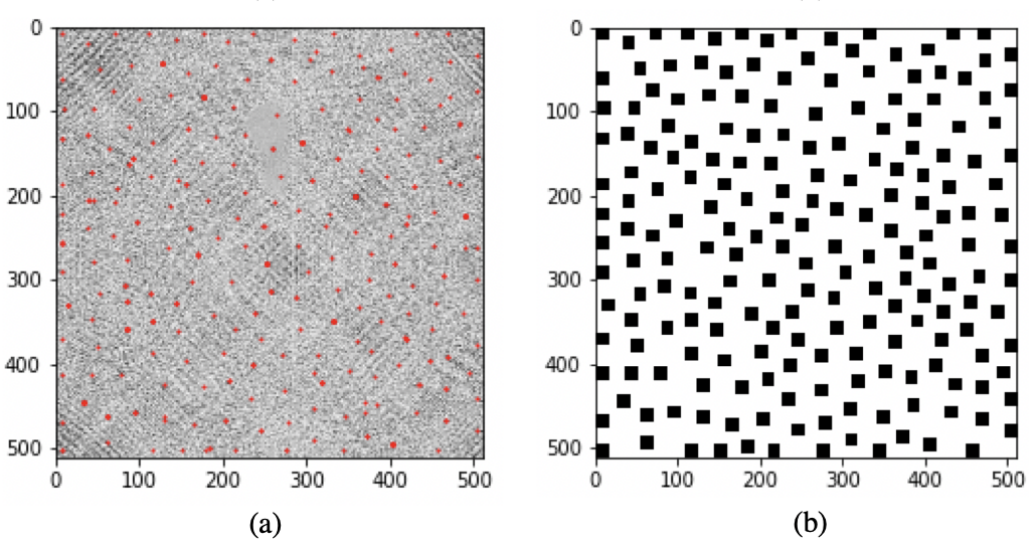

Figure 4.13: Sampling points and patches pattern obtained after applying maximin sampling

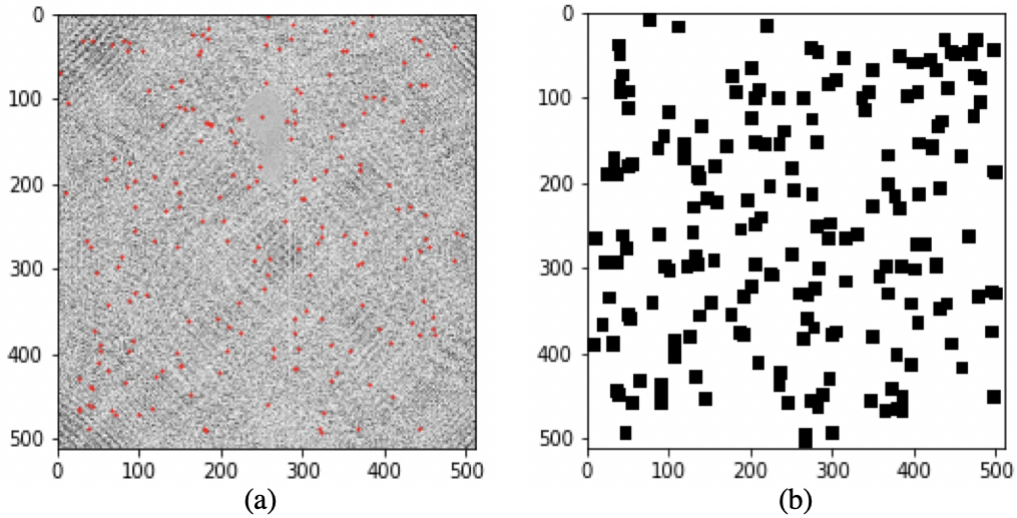

Figure 4.14: Sampling points and patches pattern obtained after applying random sampling

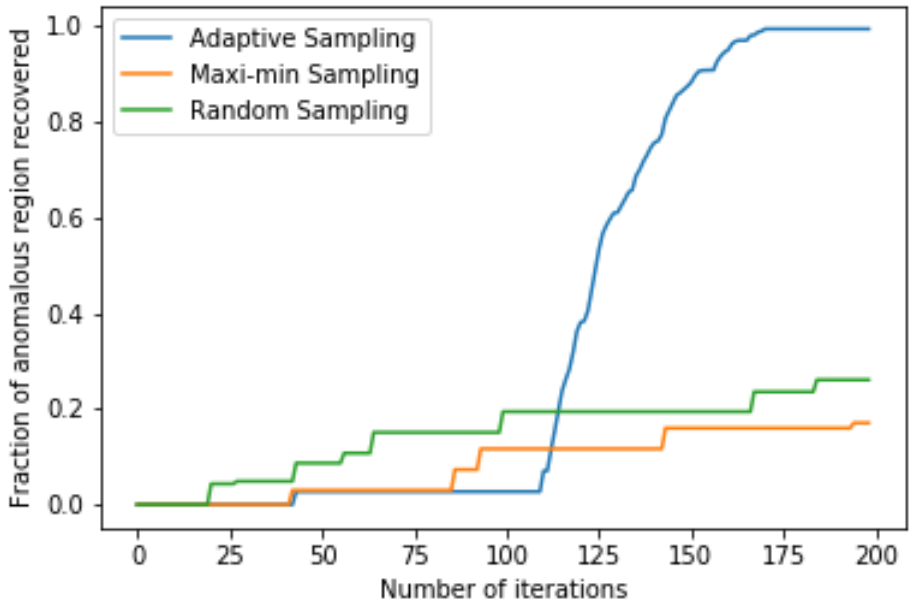

Figure 4.15: Fraction of anomaly detected using different methods as a function of sampling iterations

4.5 Conclusion

Feature detection in videos and images is an important task encountered in various applications such as evaluating a candidate using an interview video and inspecting a product using an image. Artificial intelligence (AI) will impact the recruitment process in a positive way by making it much more objective. AI-powered automated video interview evaluation can help the human resource department of any company by reducing the workforce and time required for evaluating candidates. In such a scenario, candidates will be asked to submit a video resume and/or appear for an online video interview. Therefore, candidates will also need to practice video interviews. Recently, an online interview practice platform has started. Such a service provides users with an opportunity to prepare for their interviews. Naturally, an evaluation after the practice interview is expected. Evaluation could be made at different levels such as technical and behavioral. An ad hoc way would be to watch the whole interview and evaluate the user. This can be both expensive and time-consuming. Although evaluation at the technical level may need a sophisticated solution, an AI-enabled approach for evaluation at the behavioral level is achievable. Apparent behavior of a candidate can be assessed based on emotions shown during the video. Also, negative emotions are crucial from the evaluation point of view. Such emotions are usually sparsely present in the video. Processing the whole video to detect such emotions takes a significant amount of time. Therefore, a smart approach for detecting such emotions without processing the entire video is needed.

Exploring areas of a large product for anomaly detection is crucial in quality control. For example, in non-destructive evaluation (NDE), a point-based sensing system is used for inspection and anomaly detection [62]. Most point-based sensing systems measure one point at a time. Also, there are patch-based sensing systems measuring a set of pixels (sub-image) at a time. But such systems are time-consuming. Furthermore, anomalies can be found that are indistinguishable from an image background because they get introduced

in a product in a way that they deviate from the background structurally rather than in terms of pixel intensities. Often, anomalies are also clustered and sparse. Therefore, a sequential and adaptive sampling strategy coupled with a classification model is needed to detect sparse and indistinguishable anomalies.

In this work, we propose an approach to adaptively sample frames from the video to detect the desired emotions. With the smartly detected emotions, the evaluation can be targeted around the selected frames. Such a technique can help reduce candidate evaluation time. The same approach is also applicable in the case of the image where anomaly detection needs to be carried out. The proposed approach reduces the amount of data to be processed or sensed. This reduces processing/sensing time. The proposed approach also enables effective detection of emotions and separation of anomalies from the background.

CHAPTER 5
CONCLUSION

5.1 Summary of Original Contributions

This thesis investigates three major research topics in the area of unlabeled high-dimensional data analysis. In Chapter 2, we consider clustering censored trajectory data. In Chapter 3, we consider detecting pixel-level features from medical imaging data. In Chapter 4, we focus on detecting sparse features from interview videos and textural surface images.

The original contribution of Chapter 2 is to propose a novel unsupervised learning framework for clustering censored trajectory data. The proposed framework exploits a mixture model-based clustering method. The mixture of semi-Markov models is considered to model the transition dynamics embedded in the data. Each mixture component, i.e., semi-Markov model, explicitly accounts for censoring. An expectation maximization-based algorithm is developed to estimate model parameters and infer cluster assignments. Using the simulation study, we evaluate the performance of the proposed framework and compare it with the existing methods. The proposed framework is also applied to a real case where the obtained clustering results are found to be useful for devising better advertisement and marketing strategies. We believe the proposed framework has a great potential to play an impactful role while carrying out customer segmentation. This method can also be applied to various censored or uncensored spatio-temporal data settings.

The original contribution of Chapter 3 is to propose a novel, automatic image decomposition-based sparse extreme pixel-level feature detection model. This model decomposes an image into four components – mean, positive extreme features, negative extreme features, and noise. A high-dimensional least squares regression is utilized to estimate model parameters. Appropriate regularization and constraints are also included in the large-scale

optimization problem. An efficient algorithm based on the alternating direction method of multipliers and the proximal gradient method is developed to solve the optimization problem. In the simulation and the real case studies, the proposed model is able to find features more precisely than the existing methods. When applied to the real case of a computed tomography image of the human heart, this model has shown the capability to help surgeons in medical treatment planning. This model can also be applied to detect anomalies in single images occurring in various manufacturing systems.

The original contribution of Chapter 4 is to propose a novel sampling approach to detect sparse features in high-dimensional input. The proposed approach utilizes an adaptive sampling technique and a convolutional neural network model. The sampling criterion explores the high-dimensional input and exploits the regions of interest. The convolutional neural network model calculates the probability of a particular region being desirable or not, which helps the exploitation component of the sampling strategy. Using artificial and real data sets, the performance of the proposed approach is demonstrated. The proposed approach is impactful as it significantly reduces the time required for detecting emotions in a candidate's interview video. Also, it has shown the potential to minimize the amount of image data to be sensed and processed while effectively identifying the anomalies.

5.2 Future Work

The unlabeled data analysis for system performance improvement is a very active research area currently. There is a scope for future work.

For the research topic discussed in Chapter 2, one direction of future work could be to apply the proposed clustering methodology to the case of user behavior analytics for other services such as Netflix. Also, the proposed clustering methodology can be adapted to patient behavior modeling and clustering based on disease progression data. In terms of methodology, other possible ways to consider the effect of censoring could be explored for better behavior modeling. Additionally, the effect of demographic attributes, such as

age, sex, and education, could be considered while modeling the behavior using the semi-Markov model.

Some future research directions for the research topic discussed in Chapter 3 are as follows. One future work opportunity could be to extend the proposed pixel-level feature detection model to the case of $3-$dimensional CT image to extract the calcification and the aortic valve morphology. As far as methodology is concerned, other possible ways to dynamically tune the ADMM hyper-parameter could be explored. Another interesting future research direction is to develop a data-driven approach to learn the appropriate basis for feature representation.

The research topic discussed in Chapter 4 also offers future research opportunities. One could try to adaptively detect audio features from interview videos. Also, other features such as posture and hand gestures could be analyzed. All these would require smart classification models. One could explore better deep learning-based models to assist the exploitation component of the adaptive sampling approach. Also, product images having multiple types of defects could utilize the proposed sampling approach to perform selective defect detection.

Appendices

APPENDIX A

SUPPLEMENTARY MATERIAL OF "SPATIO-TEMPORAL CLUSTERING FOR CENSORED TRAJECTORY DATA"

A.1 Mixture Weights Estimation

To estimate mixture weights, the following optimization problem needs to be solved:

$$\begin{aligned} &\underset{\pi^{(1)},\ldots,\pi^{(K')}}{\text{argmin}} \quad \sum_{l=1}^{L}\sum_{k'\in K'} \Omega_{lk'}(\boldsymbol{\Theta}^{(p)})\ \log\ \pi^{(k')} + \alpha \sum_{l=1}^{L}\sum_{k'=1}^{K'} \pi^{(k')}\ \log\ \pi^{(k')} \\ &\text{subject to} \quad \sum_{k'=1}^{K'} \pi^{k'} = 1. \end{aligned} \tag{A.1}$$

To carry out this constrained optimization problem, Lagrange multiplier λ_π is introduced and the Lagrangian is written as follows:

$$\begin{aligned} L(\lambda_\pi, \pi^{(1)}, ..., \pi^{(K')}|\boldsymbol{\Theta}^{(p)}) \ &= \ \sum_{l=1}^{L}\sum_{k'\in K'} \Omega_{lk'}(\boldsymbol{\Theta}^{(p)})\ \log\ \pi^{(k')} + \alpha \sum_{l=1}^{L}\sum_{k'=1}^{K'} \pi^{(k')}\ \log\ \pi^{(k')} \\ &\quad \lambda_\pi(\sum_{k'\in K'} \pi^{(k')} - 1). \end{aligned} \tag{A.2}$$

Next, we take the derivatives of the A.2 with respect to $\pi^{(1)}, ..., \pi^{(K')}$, set them to zero, and obtain the following:

$$
\begin{aligned}
\frac{\partial L}{\partial \pi^{(k')}} &= \sum_{l=1}^{L} \Omega_{lk'}(\boldsymbol{\Theta}^{(p)}) \frac{1}{\pi^{(k')}} + \alpha \sum_{l=1}^{L} (\log\, \pi^{(k')} + 1) + \lambda_\pi \\
0 &= \sum_{l=1}^{L} \Omega_{lk'}(\boldsymbol{\Theta}^{(p)}) \frac{1}{\pi^{(k')}} + \alpha L (\log\, \pi^{(k')} + 1) + \lambda_\pi \\
0 &= \sum_{l=1}^{L} \Omega_{lk'}(\boldsymbol{\Theta}^{(p)}) + \alpha L \pi^{(k')} (\log\, \pi^{(k')} + 1) + \lambda_\pi \pi^{(k')}. \qquad \text{(A.3)}
\end{aligned}
$$

Summing over $k^{'}$ each side of A.3, we get:

$$
0 = \sum_{k' \in K'} \sum_{l=1}^{L} \Omega_{lk'}(\boldsymbol{\Theta}^{(p)}) + \alpha L \sum_{k' \in K'} \pi^{(k')} (\log\, \pi^{(k')} + 1) + \lambda_\pi \sum_{k' \in K'} \pi^{(k')}.
$$

Solving for λ_π, it gives:

$$
\begin{aligned}
\lambda_\pi &= -\sum_{k' \in K'} \sum_{l=1}^{L} \Omega_{lk'}(\boldsymbol{\Theta}^{(p)}) - \alpha L \sum_{k' \in K'} \pi^{(k')} (\log\, \pi^{(k')} + 1) \\
&= -L - \alpha L \sum_{k' \in K'} \pi^{(k')} (\log\, \pi^{(k')} + 1). \qquad \text{(A.4)}
\end{aligned}
$$

Substituting A.4 in A.3 and simplifying, we obtain:

$$
\begin{aligned}
0 &= \sum_{l=1}^{L} \Omega_{lk'}(\boldsymbol{\Theta}^{(p)}) + \alpha L \pi^{(k')} (\log\, \pi^{(k')} + 1) - \pi^{(k')} L - \alpha \pi^{(k')} L \sum_{k' \in K'} \pi^{(k')} (\log\, \pi^{(k')} + 1) \\
0 &= \sum_{l=1}^{L} \Omega_{lk'}(\boldsymbol{\Theta}^{(p)}) + \alpha L \pi^{(k')} \log\, \pi^{(k')} + \alpha L \pi^{(k')} - \pi^{(k')} L - \alpha \pi^{(k')} L \sum_{k' \in K'} \pi^{(k')} \log\, \pi^{(k')} \\
&\quad - \alpha \pi^{(k')} L \sum_{k' \in K'} \pi^{(k')} \\
0 &= \sum_{l=1}^{L} \Omega_{lk'}(\boldsymbol{\Theta}^{(p)}) + \alpha L \pi^{(k')} \log\, \pi^{(k')} + \alpha L \pi^{(k')} - \pi^{(k')} L - \alpha \pi^{(k')} L \sum_{k' \in K'} \pi^{(k')} \log\, \pi^{(k')} \\
&\quad - \alpha \pi^{(k')} L \\
0 &= \sum_{l=1}^{L} \Omega_{lk'}(\boldsymbol{\Theta}^{(p)}) + \alpha L \pi^{(k')} \log\, \pi^{(k')} - \pi^{(k')} L - \alpha \pi^{(k')} L \sum_{k' \in K'} \pi^{(k')} \log\, \pi^{(k')}.
\end{aligned}
$$

Therefore, the update equation for the mixture weights $\pi^{k^{'}}$ can be written as follows:

$$\pi_{new}^{(k')} = \frac{\sum_{l=1}^{L} \Omega_{lk'}(\boldsymbol{\Theta}^{(p)})}{L} + \alpha\pi_{old}^{(k')}(\log\ \pi_{old}^{(k')} - \sum_{k' \in K'} \pi_{old}^{(k')} \log\ \pi_{old}^{(k')}). \tag{A.5}$$

A.2 Semi-Markov Models Parameters Estimation

To estimate mixture components parameters, the following optimization problem needs to be solved:

$$\begin{aligned} &\underset{\{\boldsymbol{\rho}^{(k')}, \boldsymbol{\lambda}^{(k')}\}_{k'=1}^{K'}}{\text{argmin}} \quad \sum_{k' \in K'} \log \left[\prod_{i=1}^{s_1} \left[(\rho_i^{(k')})^{w_i^{(k')}} \prod_{k=1}^{M} \left\{ (q_{ik}^{(k')})^{(n_{ik}^{(k')} - d_{ik}^{(k')})} \prod_{j=1}^{s} (\lambda_{ijk}^{(k')})^{m_{ijk}^{(k')}} \right\} \right] \right] \\ &\text{subject to} \qquad \sum_{i=1}^{s_1} \rho_i^{(k')} = 1, \forall k' = 1, ..., K^{'}. \end{aligned} \tag{A.6}$$

To carry out this constrained optimization problem, Lagrange multipliers $\{\lambda_{\rho_{k'}}\}_{k'=1}^{K^{'}}$ are introduced and the Lagrangian is written as follows:

$$\begin{aligned} L(\{\lambda_{\rho_{k'}}, \boldsymbol{\rho}^{(k^{'})}, \boldsymbol{\lambda}^{(k^{'})}\}_{k'=1}^{K^{'}} | \boldsymbol{\Theta}^{(p)}) &= \sum_{k' \in K'} \log \left\{ \prod_{i=1}^{s_1} (\rho_i^{(k')})^{w_i^{(k')}} \right\} + \sum_{k' \in K'} \lambda_{\rho_{k'}} (\sum_{i=1}^{s_1} \rho_i^{(k')} - 1) \\ &\quad + \sum_{k' \in K'} \log \left[\prod_{i=1}^{s_1} \left[\prod_{k=1}^{M} \left\{ (1 - \sum_{j=1}^{s} \lambda_{ijk}^{(k')})^{(n_{ik}^{(k')} - d_{ik}^{(k')})} \times \right. \right. \right. \\ &\quad \left. \left. \left. \prod_{j=1}^{s} (\lambda_{ijk}^{(k')})^{m_{ijk}^{(k')}} \right\} \right] \right] \\ &= \sum_{k' \in K'} \log \left\{ \prod_{i=1}^{s_1} (\rho_i^{(k')})^{w_i^{(k')}} \right\} + \sum_{k' \in K'} \lambda_{\rho_{k'}} (\sum_{i=1}^{s_1} \rho_i^{(k')} - 1) \\ &\quad + \sum_{k' \in K'} \sum_{i=1}^{s_1} \sum_{k=1}^{M} \left\{ \log (1 - \sum_{j=1}^{s} \lambda_{ijk}^{(k')})^{(n_{ik}^{(k')} - d_{ik}^{(k')})} \right. \\ &\quad \left. + \sum_{j=1}^{s} \log (\lambda_{ijk}^{(k')})^{m_{ijk}^{(k')}} \right\}. \end{aligned} \tag{A.7}$$

Next, we perform first derivative of the L with respect to $\rho_i^{(k')}$ and set it to zero:

$$\begin{aligned} \frac{\partial L}{\partial \rho_i^{(k')}} &= \frac{w_i^{(k')}}{\rho_i^{(k')}} + \lambda_{\rho_{k'}} \\ 0 &= \frac{w_i^{(k')}}{\rho_i^{(k')}} + \lambda_{\rho_{k'}} \\ \rho_i^{(k')} &= -\frac{w_i^{(k')}}{\lambda_{\rho_{k'}}}. \end{aligned} \tag{A.8}$$

Summing over i each side of A.8, we obtain:

$$\begin{aligned} \sum_{i=1}^{s_1} \rho_i^{(k')} &= \sum_{i=1}^{s_1} \left(-\frac{w_i^{(k')}}{\lambda_{\rho_{k'}}} \right) \\ 1 &= \sum_{i=1}^{s_1} \left(-\frac{w_i^{(k')}}{\lambda_{\rho_{k'}}} \right) \\ \lambda_{\rho_{k'}} &= -\sum_{i=1}^{s_1} w_i^{(k')}. \end{aligned} \tag{A.9}$$

Plugging A.9 into A.8, we get the following estimator:

$$\rho_{i,npmle}^{(k')} = \frac{w_i^{(k')}}{\sum_{i=1}^{s_1} w_i^{(k')}}. \tag{A.10}$$

Next, we perform first derivative of the L with respect to $\lambda_{ijk}^{(k')}$, set it to zero, and simplify

as follows:

$$\begin{aligned}
\frac{\partial L(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{(p)})}{\partial \lambda_{ijk}^{(k')}} &= -\frac{(n_{ik}^{(k')} - d_{ik}^{(k')})}{1 - \sum_{j=1}^{s} \lambda_{ijk}^{(k')}} + \frac{m_{ijk}^{(k')}}{\lambda_{ijk}^{(k')}} \\
0 &= -\frac{(n_{ik}^{(k')} - d_{ik}^{(k')})}{1 - \sum_{j=1}^{s} \lambda_{ijk}^{(k')}} + \frac{m_{ijk}^{(k')}}{\lambda_{ijk}^{(k')}} \\
\frac{\lambda_{ijk}^{(k')}}{1 - \sum_{j=1}^{s} \lambda_{ijk}^{(k')}} &= \frac{m_{ijk}^{(k')}}{(n_{ik}^{(k')} - d_{ik}^{(k')})} \\
\frac{\lambda_{ijk}^{(k')}}{1 - \sum_{j=1}^{s} \lambda_{ijk}^{(k')}} &= \frac{\frac{m_{ijk}^{(k')}}{n_{ik}^{(k')}}}{(1 - \frac{d_{ik}^{(k')}}{n_{ik}^{(k')}})} \\
\frac{\lambda_{ijk}^{(k')}}{1 - \sum_{j=1}^{s} \lambda_{ijk}^{(k')}} &= \frac{\frac{m_{ijk}^{(k')}}{n_{ik}^{(k')}}}{(1 - \sum_{j=1}^{s} \frac{m_{ijk}^{(k')}}{n_{ik}^{(k')}})}.
\end{aligned}$$

Therefore, the estimator is given by:

$$\lambda_{ijk,npmle}^{(k')} = \frac{m_{ijk}^{(k')}}{n_{ik}^{(k')}}. \tag{A.11}$$

APPENDIX B

SUPPLEMENTARY MATERIAL OF "IMAGE DECOMPOSITION-BASED SPARSE EXTREME PIXEL-LEVEL FEATURE DETECTION MODEL"

B.1 Constrained Weighted LASSO Problem

The PFD problem in 3.3 is equivalent to a constrained weighted LASSO problem in the form of:

$$\begin{aligned} \underset{\theta_p,\theta_n}{\text{argmin}} \quad & (\tilde{y} - B_p\theta_p - B_n\theta_n)^T(I-H)(\tilde{y} - B_p\theta_p - B_n\theta_n) + \gamma_p||\theta_p||_1 + \gamma_n||\theta_n||_1 \\ \text{subject to} \quad & \theta_p > 0 \\ & \theta_n < 0, \end{aligned} \tag{B.1}$$

where $H = B(B^TB + \lambda R)^{-1}B^T$.

Explanation: The PFD problem in 3.3 is first solved for θ while keeping θ_p & θ_n as fixed via following unconstrained optimization problem:

$$\underset{\theta}{\text{argmin}} \quad ||\tilde{y} - B\theta - B_p\theta_p - B_n\theta_n||^2 + \lambda\theta^T R\theta, \tag{B.2}$$

which gives, $\hat{\theta} = (B^TB + \lambda R)^{-1}B^T(\tilde{y} - B_p\theta_p - B_n\theta_n)$. So, we can write $B\hat{\theta} = B(B^TB + \lambda R)^{-1}B^T(\tilde{y} - B_p\theta_p - B_n\theta_n) = H(\tilde{y} - B_p\theta_p - B_n\theta_n)$. Next, we plug this into

3.3, we have the following:

$$
\begin{aligned}
\underset{\theta_p,\theta_n}{\text{argmin}} \quad & ||\tilde{y} - H(\tilde{y} - B_p\theta_p - B_n\theta_n) - B_p\theta_p - B_n\theta_n||^2 \\
& + \lambda((B^TB + \lambda R)^{-1}B^T(\tilde{y} - B_p\theta_p - B_n\theta_n))^T R(B^TB + \lambda R)^{-1}B^T(\tilde{y} - B_p\theta_p - B_n\theta_n) \\
& + \gamma_p||\theta_p||_1 \\
& + \gamma_n||\theta_n||_1 \\
\text{subject to} \quad & \theta_p > 0 \\
& \theta_n < 0.
\end{aligned}
\tag{B.3}
$$

Next, we simplify the first two terms in the objective function of the above problem (B.3) as follows:

$||\tilde{y} - H(\tilde{y} - B_p\theta_p - B_n\theta_n) - B_p\theta_p - B_n\theta_n||^2 + \lambda((B^TB + \lambda R)^{-1}B^T(\tilde{y} - B_p\theta_p - B_n\theta_n))^T R(B^TB + \lambda R)^{-1}B^T(\tilde{y} - B_p\theta_p - B_n\theta_n)$

$= ||(I - H)(\tilde{y} - B_p\theta_p - B_n\theta_n)||^2 + \lambda(\tilde{y} - B_p\theta_p - B_n\theta_n)^T B(B^TB + \lambda R^T)^{-1}R(B^TB + \lambda R)^{-1}B^T(\tilde{y} - B_p\theta_p - B_n\theta_n)$.

Assuming R to be a symmetric matrix and denoting $K_\lambda = (B^TB + \lambda R)$, it again simplifies as follows:

$= ||(I - H)(\tilde{y} - B_p\theta_p - B_n\theta_n)||^2 + \lambda(\tilde{y} - B_p\theta_p - B_n\theta_n)^T BK_\lambda^{-1}RK_\lambda^{-1}B^T(\tilde{y} - B_p\theta_p - B_n\theta_n)$

$= (\tilde{y} - B_p\theta_p - B_n\theta_n)^T(I - H)^T(I - H)(\tilde{y} - B_p\theta_p - B_n\theta_n)$

$+\lambda(\tilde{y} - B_p\theta_p - B_n\theta_n)^T BK_\lambda^{-1}RK_\lambda^{-1}B^T(\tilde{y} - B_p\theta_p - B_n\theta_n)$

$= (\tilde{y} - B_p\theta_p - B_n\theta_n)^T(I + H^2 - 2IH)(\tilde{y} - B_p\theta_p - B_n\theta_n)$

$+\lambda(\tilde{y} - B_p\theta_p - B_n\theta_n)^T BK_\lambda^{-1}RK_\lambda^{-1}B^T(\tilde{y} - B_p\theta_p - B_n\theta_n)$

$= (\tilde{y} - B_p\theta_p - B_n\theta_n)^T(I + BK_\lambda^{-1}B^TBK_\lambda^{-1}B^T - 2BK_\lambda^{-1}B^T)(\tilde{y} - B_p\theta_p - B_n\theta_n) + \lambda(\tilde{y} - B_p\theta_p - B_n\theta_n)^T BK_\lambda^{-1}RK_\lambda^{-1}B^T(\tilde{y} - B_p\theta_p - B_n\theta_n)$

$= (\tilde{y} - B_p\theta_p - B_n\theta_n)^T(I + BK_\lambda^{-1}B^TBK_\lambda^{-1}B^T - 2BK_\lambda^{-1}B^T)(\tilde{y} - B_p\theta_p - B_n\theta_n) + (\tilde{y} - B_p\theta_p - B_n\theta_n)^T BK_\lambda^{-1}\lambda RK_\lambda^{-1}B^T(\tilde{y} - B_p\theta_p - B_n\theta_n)$

$= (\tilde{y} - B_p\theta_p - B_n\theta_n)^T(I + BK_\lambda^{-1}B^TBK_\lambda^{-1}B^T - 2BK_\lambda^{-1}B^T + BK_\lambda^{-1}\lambda RK_\lambda^{-1}B^T)(\tilde{y} - B_p\theta_p - B_n\theta_n)$

$= (\tilde{y} - B_p\theta_p - B_n\theta_n)^T(I + BK_\lambda^{-1}(B^TB + \lambda R)K_\lambda^{-1}B^T - 2BK_\lambda^{-1}B^T)(\tilde{y} - B_p\theta_p - B_n\theta_n)$

$= (\tilde{y} - B_p\theta_p - B_n\theta_n)^T(I + BK_\lambda^{-1}K_\lambda K_\lambda^{-1}B^T - 2BK_\lambda^{-1}B^T)(\tilde{y} - B_p\theta_p - B_n\theta_n)$

$= (\tilde{y} - B_p\theta_p - B_n\theta_n)^T(I - BK_\lambda^{-1}B^T)(\tilde{y} - B_p\theta_p - B_n\theta_n)$

$= (\tilde{y} - B_p\theta_p - B_n\theta_n)^T(I - H)(\tilde{y} - B_p\theta_p - B_n\theta_n).$

Substituting this result into B.3, we get the revised optimization problem for estimating θ_p & θ_n as claimed in B.1. During the above simplification process, we discover that following relation holds true:

$$(I - H) = (I - H)^T(I - H) + BK_\lambda^{-1}\lambda RK_\lambda^{-1}B^T. \tag{B.4}$$

The relationship B.4 is also claimed to be true in Proposition 3 of [26]. Since it is not explicitly explained there (may be due to lack of space), we provide the detailed proof for the sake of completeness and reader's understanding.

B.2 ADMM Assumption

The functions $f(\theta_p, \theta_n)$ and $g(z_p, z_n)$ are closed, proper, and convex.

Explanation: $f(\theta_p, \theta_n)$ can be regarded as consisting of three parts: $(\tilde{y} - B_p\theta_p - B_n\theta_n)^T(I - H)(\tilde{y} - B_p\theta_p - B_n\theta_n)$, $\gamma_p||\theta_p||_1$, and $\gamma_n||\theta_n||_1$. The first part will be convex if $(I - H)$ can be showed to be a positive semi-definite. Assuming R to be a positive semi-definite matrix, [26] proved that $(I - H)$ is positive semi-definite using B.4. Hence, the first part is convex. Second part and third part are L_1 norms. Since L_1 norm can easily be proven to be convex, second part and third part become convex. Following the fact that linear combinations of convex functions is also convex, therefore $f(\theta_p, \theta_n)$ can be claimed to be convex because it is a linear combination of convex functions.

Since $f(\theta_p, \theta_n)$ is a convex function taking values in the extended real number line such

that $f(\theta_p, \theta_n) < \infty$ for at least one (θ_p, θ_n) and $f(\theta_p, \theta_n) > -\infty$ for every (θ_p, θ_n), it can also be claimed to be proper.

A proper convex function is closed iff it is lower semi-continuous. Since $f(\theta_p, \theta_n)$ consists of (positively) weighted sum of continuous functions, it is indeed semi-continuous. To see whether $g(z_p, z_n) = \begin{cases} 0 & \text{if } (z_p, z_n) \in \mathcal{C} \\ \infty & \text{if } (z_p, z_n) \notin \mathcal{C} \end{cases}$, where $\mathcal{C} = \{(z_p, z_n) : z_p > 0, z_n < 0\}$ is convex, it needs to be checked whether set $\mathcal{C}$ is convex. Suppose $\nu = (\nu_p, \nu_n) \in \mathcal{C}$ and $\upsilon = (\upsilon_p, \upsilon_n) \in \mathcal{C}$. Then, $\forall \psi \in \mathbb{R}$ and $0 \leq \psi \leq 1$, and $\forall \nu, \upsilon \in \mathcal{C}$, the following holds: $\psi(\nu_p, \nu_n) + (1 - \psi)(\upsilon_p, \upsilon_n) \in \mathcal{C}$. Therefore, set $\mathcal{C}$ is convex. Hence, $g(z_p, z_n)$ is convex.

It is easy to see $g(z_p, z_n)$ is proper because $g(z_p, z_n)$ is a convex function taking values in the extended real number line such that $g(z_p, z_n) < \infty$ for at least one (z_p, z_n) and $g(z_p, z_n) > -\infty$ for every (z_p, z_n),

To see whether proper convex function $g(z_p, z_n)$ is closed, it should be checked whether $g(z_p, z_n)$ is lower semi-continuous. Consider a point $\nu = (\nu_p, \nu_n) \in \mathcal{C}$ for which we have $g(\nu) = 0$. The function values for arguments near ν are also 0. Consider a point $\nu = (0 - \epsilon, 0 + \epsilon)$, where $\epsilon > 0$ is a very small number. The function values in this neighborhood is ∞ which is greater than the function value at $\nu = (0+\epsilon, 0-\epsilon)$. Therefore, it can be said that $g(z_p, z_n)$ is lower semi-continuous and hence, is closed proper convex function.

B.3 PG Method Assumption

Proof of Proposition 1: Gradient of $F(\theta_p)$ can be calculated as shown below:

$$\nabla F(\theta_p) = 2B_p^T(I - H)(B_p\theta_p + B_n\theta_n^{(k-1)} - \tilde{y}) + \rho(\theta_p - z_p^{(k-1)} + u_p^{(k-1)}). \tag{B.5}$$

Note that $||X||_2$ represents spectral norm of matrix X. Next, we calculate the following

quantity:

$$\begin{aligned} ||\nabla F(\alpha) - \nabla F(\beta)|| &= ||2B_p^T(I-H)B_p(\alpha-\beta) + \rho(\alpha-\beta)|| \\ &\leq ||2B_p^T(I-H)B_p(\alpha-\beta)|| + \rho \cdot ||(\alpha-\beta)|| \\ &\leq ||2B_p^T(I-H)B_p||_2 \cdot ||(\alpha-\beta)|| + \rho \cdot ||(\alpha-\beta)|| \\ &\leq ||2B_p^T||_2 \cdot ||(I-H)||_2 \cdot ||B_p||_2 \cdot ||(\alpha-\beta)|| + \rho \cdot ||(\alpha-\beta)||. \end{aligned} \tag{B.6}$$

Using the result $||I-H||_2 \leq 1$ from claim 5 in [26], it further simplifies as follows:

$$\begin{aligned} ||\nabla F(\alpha) - \nabla F(\beta)|| &\leq ||2B_p^T||_2 \cdot ||B_p||_2 \cdot ||(\alpha-\beta)|| + \rho \cdot ||(\alpha-\beta)|| \\ &\leq (2||B_p||_2^2 + \rho)||(\alpha-\beta)||. \end{aligned} \tag{B.7}$$

It is easy to deduce that $L = 2||B_p||_2^2 + \rho$.

B.4 PG Method Closed-form Solution

Proof Proposition 2: We are required to solve the following problem:

$$\underset{\theta_p}{\text{argmin}} \quad F(\theta_p^{(k-1)}) + \langle \theta_p - \theta_p^{(k-1)}, \nabla F(\theta_p^{(k-1)}) \rangle + \frac{L}{2}||\theta_p - \theta_p^{(k-1)}||^2 + \gamma_p||\theta_p||_1. \tag{B.8}$$

First, we simplify the objective function in the following fashion:

$F(\theta_p^{(k-1)}) + \langle \theta_p - \theta_p^{(k-1)}, \nabla F(\theta_p^{(k-1)}) \rangle + \frac{L}{2}||\theta_p - \theta_p^{(k-1)}||^2 + \gamma_p||\theta_p||_1$

$= (\theta_p - \theta_p^{(k-1)})^T(2B_p^T(I-H)(B_p\theta_p^{(k-1)} + B_n\theta_n^{(k-1)} - \tilde{y}) + \rho(\theta_p^{(k-1)} - z_p^{(k-1)} + u_p^{(k-1)}))$

$+ \frac{L}{2}||\theta_p - \theta_p^{(k-1)}||^2 + \gamma_p||\theta_p||_1.$

Multiplying by $\frac{2}{L}$,

$= 2(\theta_p - \theta_p^{(k-1)})^T(\frac{2}{L}B_p^T(I-H)(B_p\theta_p^{(k-1)} + B_n\theta_n^{(k-1)} - \tilde{y}) + \rho(\theta_p^{(k-1)} - z_p^{(k-1)} + u_p^{(k-1)}))$

$+ ||\theta_p - \theta_p^{(k-1)}||^2 + \frac{2}{L}\gamma_p||\theta_p||_1.$

Adding a constant term to make a perfect square,

$= ||\theta_p - \theta_p^{(k-1)} + \frac{2}{L}B_p^T(I-H)(B_p\theta_p^{(k-1)} + B_n\theta_n^{(k-1)} - \tilde{y}) + \rho(\theta_p^{(k-1)} - z_p^{(k-1)} + u_p^{(k-1)})||^2 +$

$\frac{2}{L}\gamma_p||\theta_p||_1$

$= ||\theta_p - \theta_p^{(k-1)} + \frac{2}{L}B_p^T(B_p\theta_p^{(k-1)} + B_n\theta_n^{(k-1)} - \tilde{y} + H(\tilde{y} - B_p\theta_p^{(k-1)} - B_n\theta_n^{(k-1)}) + \rho(\theta_p^{(k-1)} - z_p^{(k-1)} + u_p^{(k-1)})||^2 + \frac{2}{L}\gamma_p||\theta_p||_1$

$= ||\theta_p - \theta_p^{(k-1)} + \frac{2}{L}B_p^T(B_p\theta_p^{(k-1)} + B_n\theta_n^{(k-1)} - \tilde{y} + B\theta^{(t-1)}) + \rho(\theta_p^{(k-1)} - z_p^{(k-1)} + u_p^{(k-1)})||^2 + \frac{2}{L}\gamma_p||\theta_p||_1$

$= ||\theta_p - \theta_p^{(k-1)} - \frac{2}{L}B_p^T(\tilde{y} - B\theta^{(t-1)} - B_p\theta_p^{(k-1)} - B_n\theta_n^{(k-1)}) + \rho(\theta_p^{(k-1)} - z_p^{(k-1)} + u_p^{(k-1)})||^2 + \frac{2}{L}\gamma_p||\theta_p||_1$.

Substituting this in B.8 and we get the revised form of the problem as shown below:

$$\begin{aligned}\underset{\theta_p}{\text{argmin}} \quad & ||\theta_p - \theta_p^{(k-1)} - \frac{2}{L}B_p^T(\tilde{y} - B\theta^{(t-1)} - B_p\theta_p^{(k-1)} - B_n\theta_n^{(k-1)}) + \\ & \rho(\theta_p^{(k-1)} - z_p^{(k-1)} + u_p^{(k-1)})||^2 + \frac{2}{L}\gamma_p||\theta_p||_1.\end{aligned} \tag{B.9}$$

It is easy to see that the solution to the problem in B.9 is given by the following soft-thresholding operator:

$$\theta_p^{(k)} = S_{\frac{\gamma_p}{L}}(\theta_p^{(k-1)} + \frac{2}{L}B_p^T(\tilde{y} - B\theta^{(t-1)} - B_p\theta_p^{(k-1)} - B_n\theta_n^{(k-1)}) - \rho(\theta_p^{(k-1)} - z_p^{(k-1)} + u_p^{(k-1)})). \tag{B.10}$$

REFERENCES

[1] C. Ranjan, K. Paynabar, J. Helm, and J. Pan, "The impact of estimation: A new method for clustering and trajectory estimation in patient flow modeling," *Production and Operations Management*, vol. 26, pp. 1893–1914, 10 2017.

[2] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning, Second Edition: Data Mining, Inference, and Prediction*. Springer Series in Statistics, Springer, 2010.

[3] J. MacQueen, "Some methods for classification and analysis of multivariate observations," *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297, 1967.

[4] J. Hartigan and M. Wong, "A k-means clustering algorithm," *Applied Statistics*, vol. 28, pp. 100–108, 1979.

[5] J. Bezdek, "Numerical taxonomy with fuzzy sets," *Journal of Mathematical Biology*, vol. A, pp. 57–71, 1974.

[6] J. Cuesta, J. Albertos, and C. Matran, "Trimmed k-means: An attempt to robustify quantizers," *Annals of Statistics*, vol. 25, pp. 553–576, 1997.

[7] L. Garcia-Escudero and A. Gordaliza, "Robustness properties of kmeans and trimmed k-means," *Journal of the American Statistical Association*, vol. 94, pp. 956–969, 1999.

[8] T. Kohonen, *Self-Organizing Maps*. Springer Series in Information Sciences, Springer, 2001.

[9] G. McLachlan and D. Peel, *Finite mixture models*. Wiley, 2000.

[10] J. Banfield and A. Raftery, "Model-based gaussian and nongaussian clustering," *Biometrics*, vol. 49, pp. 803–821, 3 1993.

[11] G. McLachlan and K. Basford, *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, New York, 1988.

[12] C. Fraley and A. Raftery, "Model-based clustering, discriminant analysis, and density estimation," *Journal of the American Statistical Association*, vol. 97, pp. 611–631, 2002.

[13] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of The Royal Statistical Society, B*, vol. 39, pp. 1–38, 1 1977.

[14] G. McLachlan and T. Krishnan, *The EM algorithm and extensions*. Wiley, 1997.

[15] G. Schwarz, "Estimating the dimension of a model," *Annals of Statistics*, vol. 6, pp. 461–464, 1978.

[16] M.-S. Yang, C.-Y. Lai, and C.-Y. Lin, "A robust em clustering algorithm for gaussian mixture models," *Pattern Recognition*, vol. 45, pp. 3950–3961, 11 2012.

[17] S. Gaffney, "Probabilistic curve-aligned clustering and prediction with regression mixture models," *Ph.D. dissertation, Department of Computer Science, University of California, Irvine*, 2004.

[18] F. Chamroukhi, "Hidden process regression for curve modeling, classification and tracking," *Ph.D. Thesis, Universit de Technologie de Compigne, Compigne, France*, 2010.

[19] A. Sam, F. Chamroukhi, G. Govaert, and P. Aknin, "Model-based clustering and segmentation of time series with changes in regime," *Advances in Data Analysis and Classification*, vol. 5, pp. 1–21, 4 2011.

[20] F. Chamroukhi, A. Sam, G. Govaert, and P. Aknin, "Time series modeling by a regression approach based on a latent process," *Neural Networks*, vol. 22, pp. 593–602, 5-6 2009.

[21] J. Dias and F. Willekens, "Model-based clustering of sequential data with an application to contraceptive use dynamics," *Mathematical Population Studies*, vol. 12, pp. 135–157, 2005.

[22] S. Lagakos, C. Sommer, and M. Zelen, "Semi-markov models for partially censored data," *Biometrika*, vol. 65, pp. 311–317, 2 1978.

[23] G. Dinse and M. Larson, "A note on semi-markov models for partially censored data," *Biometrika*, vol. 73, pp. 379–386, 2 1986.

[24] E. Kaplan and P. Meier, "Nonparametric estimation from incomplete observations," *Journal of the American Statistical Association*, vol. 53, pp. 457–481, 1958.

[25] F. Chamroukhi, "Unsupervised learning of regression mixture models with unknown number of components," *Journal of Statistical Computation and Simulation*, vol. 86, pp. 2308–2334, 12 2016.

[26] H. Yan, K. Paynabar, and J. Shi, "Anomaly detection in images with smooth background via smooth-sparse decomposition," *Technometrics*, vol. 59, pp. 102–114, 1 2017.

[27] Z. Qian, L. S. Wang K. and, X. Zhou, V. Rajagopal, K. J. Meduri C. and, Y. Chang, C. Wu, C Zhang, B. Wang, and M. Vannan, "Quantitative prediction of paravalvular leak in transcatheter aortic valve replacement based on tissue-mimicking 3d printing," *JACC Cardiovasc Imaging*, vol. 10, pp. 719–731, 7 2017.

[28] Z. H. Wang, G. Lahoti, K. Wang, S. Liu, C. Zhang, B. Wang, C.-W. Wu, M. Vannan, and Z. Qian, "Prediction of paravalvular leak post transcatheter aortic valve replacement using a convolutional neural network," in *Biomedical Imaging (ISBI 2018), 2018 IEEE 15th International Symposium on*, IEEE, 2018, pp. 1088–1091.

[29] J. Chen, K. Wang, C. Zhang, and B. Wang, "An efficient statistical approach to design 3D-printed metamaterials for mimicking mechanical properties of soft biological tissues," *Additive Manufacturing*, vol. 24, pp. 341–352, 2018.

[30] J. Chen, Y. Xie, K. Wang, Z. H. Wang, G. Lahoti, C. Zhang, M. A. Vannan, B. Wang, and Z. Qian, "Generative invertible networks (GIN): Pathophysiology-interpretable feature mapping and virtual patient generation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2018, pp. 537–545.

[31] J. Chen, Y. Xie, K. Wang, C. Zhang, M. A. Vannan, B. Wang, and Z. Qian, "Avp: Physics-informed data generation for small-data learning," *arXiv preprint arXiv:1902.01522*, 2019.

[32] V Tuncay, N Prakken, P. van Ooijen, R. Budde, T Leiner, and M Oudkerk, "Semi-automatic, quantitative measurement of aortic valve area using cta: Validation and comparison with transthoracic echocardiography," *BioMed Research International*, vol. 2015, 2015.

[33] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, pp. 62–66, 1 1979.

[34] P. Sahoo, S. Soltani, and A. Wong, "A survey of thresholding techniques," *Computer Vision, Graphics, and Image Processing*, vol. 41, pp. 233–260, 1988.

[35] N. Sharma and L. Aggarwal, "Automated medical image segmentation techniques," *Journal of Medical Physics*, vol. 35, pp. 3–14, 2010.

[36] R. Adams and L. Bischof, "Seeded region growing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 641–647, 1994.

[37] F. Leymarie and M. D. Levine, "Tracking deformable objects in the plane using an active contour model," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 6, pp. 617–634, 1993.

[38] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[39] C. Xu and J. L. Prince, "Snakes, shapes, and gradient vector flow," *IEEE Transactions on image processing*, vol. 7, no. 3, pp. 359–369, 1998.

[40] N. Sharma and L. M. Aggarwal, "Automated medical image segmentation techniques," *Journal of medical physics/Association of Medical Physicists of India*, vol. 35, no. 1, p. 3, 2010.

[41] P. H. Kitslaar, R. vant Klooster, M. Staring, B. P. Lelieveldt, and R. J. van der Geest, "Segmentation of branching vascular structures using adaptive subdivision surface fitting," in *Medical Imaging 2015: Image Processing*, International Society for Optics and Photonics, vol. 9413, 2015, 94133Z.

[42] X. Gao, P. H. Kitslaar, A. J. Scholte, B. P. Lelieveldt, J. Dijkstra, and J. H. Reiber, "Automatic aortic root segmentation in cta whole-body dataset," in *Medical Imaging 2016: Computer-Aided Diagnosis*, International Society for Optics and Photonics, vol. 9785, 2016, 97850F.

[43] T. Cootes, C. Taylor, D. Cooper, and J. Graham, "Active shape models-their training and application," *Computer Vision and Image Understanding*, vol. 61, pp. 38–59, 1995.

[44] C. Wang, N. Komodakis, and N. Paragios, "Markov random field modeling, inference & learning in computer vision & image understanding: A survey," *Computer Vision and Image Understanding*, vol. 117, pp. 1610–1627, 2013.

[45] C. Nieuwenhuis, E. Toppe, and D. Cremers, "A survey and comparison of discrete and continuous multi-label optimization approaches for the potts model," *International Journal of Computer Vision*, vol. 104, pp. 223–240, 2013.

[46] M. S. Nosrati and G. Hamarneh, "Incorporating prior knowledge in medical image segmentation: A survey," *arXiv preprint arXiv:1607.01092*, 2016.

[47] Y. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Prgramming*. Society for Industrial and Applied Mathematics, 1994.

[48] H. Zou, "The adaptive lasso and its oracle properties," *Journal of the American Statistical Association*, vol. 101, pp. 1418–1429, 476 2006.

[49] B. R. Gaines, J. Kim, and H. Zhou, "Algorithms for fitting the constrained lasso," *Journal of Computational and Graphical Statistics*, vol. 27, pp. 861–871, 4 2018.

[50] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, pp. 1–122, 1 2010.

[51] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *The Annals of Statistics*, vol. 32, pp. 407–499, 2 2004.

[52] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society, Series B*, vol. 58, pp. 267–288, 1 1996.

[53] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, pp. 123–231, 3 2013.

[54] P. Tseng, "On accelerated proximal gradient methods for convex-concave optimization," *SIAM Journal on Optimization*, 2008.

[55] L. Xiao, Y. Li, and D. Ruppert, "Fast bivariate p-splines:the sandwich smoother," *Journal of the Royal Statistical Society, Series B*, vol. 75, pp. 577–599, 3 2013.

[56] P. Eilers and B. Marx, "Flexible smoothing with b-splines and penalties," *Statistical Science*, vol. 11, pp. 89–102, 2 1996.

[57] D. Ruppert, "Selecting the number of knots for penalized splines," *Journal of Computational and Graphical Statistics*, vol. 11, pp. 735–757, 4 2002.

[58] J. Huang, Z. Qian, X. Huang, D. Metaxas, and L. Axel, "Tag separation in cardiac tagged mri," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2008, pp. 289–297.

[59] H. Yan, K. Paynabar, and J. Shi, "Real-time monitoring of high-dimensional functional data streams via spatio-temporal smooth sparse decomposition," *Technometrics*, vol. 60, no. 2, pp. 181–197, 2018.

[60] L. S. Nguyen and D. Gatica-Perez, "Hirability in the wild: Analysis of online conversational video resumes," *IEEE Transactions on Multimedia*, vol. 18, no. 7, pp. 1422–1437, 2016.

[61] *Weakly supervised learning for industrial optical inspection*, `https://hci.iwr.uni-heidelberg.de/node/3616`, Accessed: 2018-10-01.

[62] H. Yan, K. Paynabar, and J. Shi, "AKM^{2}D: An adaptive framework for online sensing and anomaly detection," *Submitted to IISE Transactions*, 2018.

[63] A. N. Finnerty, S. Muralidhar, L. S. Nguyen, F. Pianesi, and D. Gatica-Perez, "Stressful first impressions in job interviews," in *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, ACM, 2016, pp. 325–332.

[64] I. Naim, M. I. Tanveer, D. Gildea, and M. E. Hoque, "Automated analysis and prediction of job interview performance," *IEEE Transactions on Affective Computing*, vol. 9, no. 2, pp. 191–204, 2018.

[65] H. Kaya, F. Gurpinar, and A. Ali Salah, "Multi-modal score fusion and decision trees for explainable automatic job candidate screening from video cvs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 1–9.

[66] Y. Güçlütürk, U. Güçlü, X. Baro, H. J. Escalante, I. Guyon, S. Escalera, M. A. Van Gerven, and R. Van Lier, "Multimodal first impression analysis with deep residual networks," *IEEE Transactions on Affective Computing*, vol. 9, no. 3, pp. 316–329, 2018.

[67] Y. Gucluturk, U. Guclu, M. Perez, H. Jair Escalante, X. Baro, I. Guyon, C. Andujar, J. Jacques Junior, M. Madadi, S. Escalera, *et al.*, "Visualizing apparent personality analysis with deep residual networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3101–3109.

[68] X. Xie, "A review of recent advances in surface defect detection using texture analysis techniques," *ELCVIA: electronic letters on computer vision and image analysis*, vol. 7, no. 3, pp. 1–22, 2008.

[69] A. Kumar, "Neural network based detection of local textile defects," *Pattern Recognition*, vol. 36, no. 7, pp. 1645–1659, 2003.

[70] X. Xie and M. Mirmehdi, "Texems: Texture exemplars for defect detection on random textured surfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 8, pp. 1454–1464, 2007.

[71] N. T. Siebel and G. Sommer, "Learning defect classifiers for visual inspection images by neuro-evolution using weakly labelled training data," in *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, IEEE, 2008, pp. 3925–3931.

[72] D Weimer, H Thamer, and B Scholz-Reiter, "Learning defect classifiers for textured surfaces using neural networks and statistical feature representations," *Procedia CIRP*, vol. 7, pp. 347–352, 2013.

[73] D. Racki, D. Tomazevic, and D. Skocaj, "A compact convolutional neural network for textured surface anomaly detection," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, 2018, pp. 1331–1339.

[74] D. Tabernik, S. Šela, J. Skvarč, and D. Skočaj, "Segmentation-based deep-learning approach for surface-defect detection," *arXiv preprint arXiv:1903.08536*, 2019.

[75] L. Qiu, X. Wu, and Z. Yu, "A high-efficiency fully convolutional networks for pixel-wise surface defect detection," *IEEE Access*, vol. 7, pp. 15 884–15 893, 2019.

[76] J. L. Loeppky, L. M. Moore, and B. J. Williams, "Batch sequential designs for computer experiments," *Journal of Statistical Planning and Inference*, vol. 140, no. 6, pp. 1452–1464, 2010.

[77] P. Ranjan, D. Bingham, and G. Michailidis, "Sequential experiment design for contour estimation from complex computer codes," *Technometrics*, vol. 50, no. 4, pp. 527–541, 2008.

[78] R. Jin, C.-J. Chang, and J. Shi, "Sequential measurement strategy for wafer geometric profile estimation," *IIE Transactions*, vol. 44, no. 1, pp. 1–12, 2012.

[79] M. E. Johnson, L. M. Moore, and D. Ylvisaker, "Minimax and maximin distance designs," *Journal of statistical planning and inference*, vol. 26, no. 2, pp. 131–148, 1990.

[80] O. Arriaga, M. Valdenegro-Toro, and P. Plöger, "Real-time convolutional neural networks for emotion and gender classification," *arXiv preprint arXiv:1710.07557*, 2017.

[81] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.